

# Northumbria Research Link

Citation: Yang, Qiangda, Fu, Yichuan and Zhang, Jie (2021) Molten steel temperature prediction using a hybrid model based on information interaction-enhanced cuckoo search. *Neural Computing and Applications*, 33 (12). pp. 6487-6509. ISSN 0941-0643

Published by: Springer

URL: <https://doi.org/10.1007/s00521-020-05413-5> <<https://doi.org/10.1007/s00521-020-05413-5>>

This version was downloaded from Northumbria Research Link:  
<http://nrl.northumbria.ac.uk/id/eprint/48049/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)



**Northumbria  
University**  
NEWCASTLE



**UniversityLibrary**



# Molten steel temperature prediction using a hybrid model based on information interaction-enhanced cuckoo search

Qiangda Yang<sup>1,2</sup> · Yichuan Fu<sup>3</sup> · Jie Zhang<sup>2</sup>

Received: 19 November 2019 / Accepted: 5 October 2020 / Published online: 23 October 2020  
© The Author(s) 2020

## Abstract

This article presents a hybrid model for predicting the temperature of molten steel in a ladle furnace (LF). Unique to the proposed hybrid prediction model is that its neural network-based empirical part is trained in an indirect way since the target outputs of this part are unavailable. A modified cuckoo search (CS) algorithm is used to optimize the parameters in the empirical part. The search of each individual in the traditional CS is normally performed independently, which may limit the algorithm's search capability. To address this, a modified CS, information interaction-enhanced CS (IICS), is proposed in this article to enhance the interaction of search information between individuals and thereby the search capability of the algorithm. The performance of the proposed IICS algorithm is first verified by testing on two benchmark sets (including 16 classical benchmark functions and 29 CEC 2017 benchmark functions) and then used in optimizing the parameters in the empirical part of the proposed hybrid prediction model. The proposed hybrid model is applied to actual production data from a 300 t LF at Baoshan Iron & Steel Co. Ltd, one of China's most famous integrated iron and steel enterprises, and the results show that the proposed hybrid prediction model is effective with comparatively high accuracy.

**Keywords** Hybrid modeling · Cuckoo search · Artificial neural networks · Molten steel temperature · Ladle furnace

## 1 Introduction

Ladle furnace (LF) is a pivotal equipment utilized to fully refine and alloy during secondary metallurgy processes in iron and steel industries [1]. Close control of the temperature of molten steel in LF is vital for the improvement of product quality and productivity [2]. However, the temperature of molten steel cannot be continuously measured in the actual production, which makes it difficult to achieve accurate control. Therefore, it has considerable practical significance to develop a model to predict the temperature of molten steel in LF.

Models for predicting the temperature of molten steel in LF are traditionally developed based on thermodynamics and the energy conservation law [3, 4]. However, due to the intrinsic complicacy of LF metallurgy processes, the fundamental mechanisms of involved physicochemical phenomena are not entirely clear by far, and developing a mechanistic prediction model is very time-consuming and costly. As a result, empirical modeling approaches have been extensively used in developing the temperature prediction models of molten steel in LF. In empirical modeling, the model is developed exclusively from the production data without the need to invoke the phenomenology of the process [5–8]. Thus, the time-consuming and expensive nature associated with the development of a suitable mechanistic prediction model can be averted.

In recent years, hybrid modeling approaches have been considered as an appealing alternative for developing molten steel temperature prediction models. A hybrid prediction model commonly consists of a mechanistic thermal model for representing the known priori knowledge of the LF metallurgy process under consideration, and

---

✉ Jie Zhang  
jie.zhang@newcastle.ac.uk

<sup>1</sup> School of Metallurgy, Northeastern University, Shenyang 110819, China

<sup>2</sup> School of Engineering, Merz Court, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

<sup>3</sup> Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne NE1 8ST, UK

one or more empirical models for approximating unknown functions in the mechanistic thermal model [9–11]. Moreover, according to existing researches, hybrid prediction models have better properties than pure empirical prediction models [9, 10]; they typically have better prediction accuracy and generalization performance, and are easier to interpret and analyze.

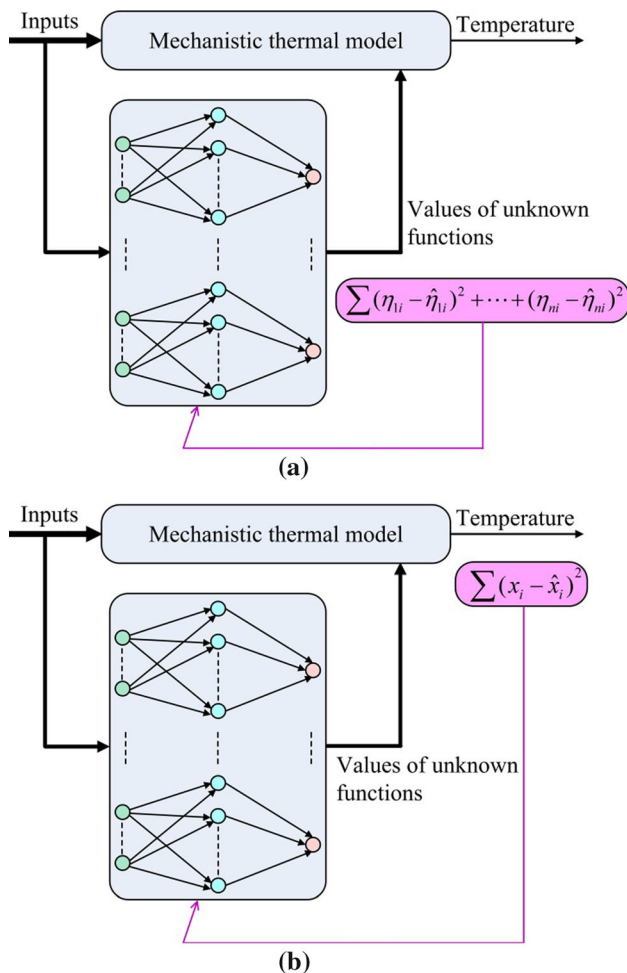
Regarding to the training of the empirical part, most of the reported hybrid modeling approaches use a direct method as schematically shown in Fig. 1a. The parameters in the empirical part are determined by minimizing the errors between outputs of the empirical part, denoted by  $\hat{\eta} = [\hat{\eta}_1, \dots, \hat{\eta}_n]$ , and the actual values of unknown functions, denoted by  $\eta = [\eta_1, \dots, \eta_n]$ . Here,  $n$  denotes the number of unknown functions. Obviously, the precondition for using the direct method to train the empirical part is that these actual values are available. In other words, when the actual values of one or more unknown functions

are unavailable just like the hybrid prediction model proposed in this article, the direct method could not be used.

To address the above issue, this article proposes an alternative method for the determination of the parameters in the empirical part using the available values of the molten steel temperature instead of the target outputs of the empirical part, as is schematically shown in Fig. 1b. This allows the empirical part being trained indirectly without having its target outputs. In Fig. 1b,  $x$  and  $\hat{x}$  denote the measured and predicted temperature values of molten steel respectively.

The determination of the parameters in the empirical part using the above indirect method is a complex optimization problem. It is difficult to calculate the derivative information required by traditional optimization algorithms. Intelligent optimization algorithms, such as the genetic algorithm (GA) [12], particle swarm optimization (PSO) [13], differential evolution (DE) [14], ant colony optimization (ACO) [15], salp swarm algorithm (SSA) [16], artificial bee colony (ABC) [17], and cuckoo search (CS) [18], do not require any derivative information and can perform global search [19–21], so using them for finding the parameters in the empirical part is a viable alternative. Amongst these algorithms, CS is a comparatively new one, initially introduced by Yang and Deb [22]. Due to some attractive features like good balance between the local search and global search, simplicity, and efficiency [23, 24], the CS algorithm has been successfully applied to many optimization problems in various fields with promising results [25–29], including the parameter optimization problems in modeling manufacturing processes such as parameter estimation of a common empirical model for the temperature of cutting tools [30], estimation of soft-sensing model parameters for fermentation processes [31], and parameter identification of a neural network model for the electron beam welding process [32]. Besides, some researches have revealed that compared with PSO, GA, and some other intelligent optimization algorithms, CS is potentially far more efficient [30, 33, 34]. However, in the search process of the basic CS (BCS), there is no interchange of search information between individuals (i.e., cuckoos). To address this issue, we propose a modified CS, information interaction-enhanced CS (IICS), by introducing an information interaction-enhanced mechanism into BCS. It is based on the common idea that the information interchange between people would be in favor of their team accomplishing an assignment with efficiency. The proposed IICS is employed to optimize the parameters in the empirical part of the proposed hybrid prediction model.

The remainder of the article is organized as follows. Section 2 elucidates the development of the hybrid temperature prediction model with the proposed indirect



**Fig. 1** Schematic representation of a hybrid molten steel temperature prediction model: **a** its empirical part is trained directly; **b** its empirical part is trained indirectly

training method for its empirical part. Section 3 briefly discusses BCS and details the IICS algorithm. Section 4 describes using IICS for determining the parameters in the empirical part of the proposed hybrid prediction model. Section 5 analyzes the performance of IICS by testing on two sets of benchmark functions and then presents the application of the proposed hybrid prediction model on the actual production data from a 300 t LF at Baoshan Iron & Steel Co. Ltd. Finally, conclusions of this study are drawn and considerations for future works are pointed out in the last section.

## 2 Development of a hybrid prediction model

In this section, a mechanistic thermal model (i.e., the mechanistic part of the proposed hybrid prediction model) is first derived based on thermodynamics as well as the law of energy conservation. Next, artificial neural network-based empirical models (i.e., the empirical part of the proposed hybrid prediction model) are used to approximate the unknown functions in the mechanistic part, and the indirect training method for these empirical models is elaborated.

### 2.1 Development of mechanistic part

Taking the molten steel and slag as a unitized system, a mechanistic thermal model is developed in this subsection based on the energy conservation law and thermodynamics. Similar to the existing literature [2], the following three assumptions are made: (1) no local temperature gradient exists in the steel bath, i.e., the steel bath is fully mixed; (2) there is only radial heat flow in the ladle wall; and (3) only axial heat flow occurs in the ladle bottom.

In the remainder of this subsection, the primary factors affecting the temperature change of molten steel are described in detail. Then, a mechanistic thermal model with two unknown functions is presented.

#### 2.1.1 Thermal gain from the arc

The energy required for secondary steel refinement in LF is mainly from the arc. The thermal gain of the steel bath due to the energy injection through arc can be calculated as

$$Q_{\text{arc}} = \eta_{\text{arc}} P_{\text{arc}} \quad (1)$$

where  $Q_{\text{arc}}$  is the power in W injected into the steel bath,  $\eta_{\text{arc}}$  is the efficiency coefficient of heat transfer from the arc to the steel bath, and  $P_{\text{arc}}$  is the total power. For a given LF system, the value of  $\eta_{\text{arc}}$  mainly depends on the slag

thickness  $H_{\text{sl}}$  and arc length  $L_{\text{arc}}$  [10]; that is to say,  $\eta_{\text{arc}}$  is a function of  $H_{\text{sl}}$  and  $L_{\text{arc}}$ , shown as

$$\eta_{\text{arc}} = f_{\text{arc}}(H_{\text{sl}}, L_{\text{arc}}) \quad (2)$$

Hence, once the function  $f_{\text{arc}}$  has been obtained, the value of  $\eta_{\text{arc}}$  can be calculated using the online available  $H_{\text{sl}}$  and  $L_{\text{arc}}$ . However, it is noteworthy that the concrete expression of this function is hard to derive by mechanistic approaches.

#### 2.1.2 Thermal loss from the ladle lining

Thermal loss from the ladle lining consists of two components, the thermal loss from the ladle wall and the thermal loss from the ladle bottom. Here, the instantaneous temperature distribution models of the ladle wall and ladle bottom are first established; then the thermal loss from the ladle lining is calculated based on these two models.

**2.1.2.1 Instantaneous temperature distribution model of the ladle wall** With the assumptions above, the heat transfer in the ladle wall can be considered as a one-dimensional unsteady heat conduction in cylindrical coordinates [2, 3, 35], formulated as

$$\frac{\partial T_w}{\partial \tau} = \frac{1}{\rho_w c_w} \left[ \frac{1}{r} \cdot \frac{\partial}{\partial r} \left( \lambda_w \cdot r \frac{\partial T_w}{\partial r} \right) \right] \quad (3)$$

$$\text{when } r = r_1, \quad T_w = T_{\text{st}} \quad (4)$$

$$\text{when } r = r_2, \quad \lambda_w \frac{\partial T_w}{\partial r} = -\alpha_{w-\text{en}}(T_w - T_{\infty}) \quad (5)$$

$$T_w(r, 0) = f(r) \quad (6)$$

Equation (3) is the heat conduction differential equation of the ladle wall, where  $T_w$ ,  $\rho_w$ ,  $\lambda_w$ , and  $c_w$  are the temperature in °C, density in kg/m<sup>3</sup>, heat conductivity in W/(m °C), and specific heat in J/(kg °C) of the ladle wall respectively. The boundary conditions for Eq. (3) are given in Eqs. (4) and (5), where  $r_1$  and  $r_2$  are respectively the inner and outer diameters of the ladle wall in m;  $T_{\text{st}}$  is the molten steel temperature in °C;  $\alpha_{w-\text{en}}$  is the convection heat transfer coefficient between the steel shell of the ladle wall and the environment in W/(m<sup>2</sup> °C); and  $T_{\infty}$  is the environment temperature in °C. Equation (6) is the initial condition. For more details, please see the existing literature [2].

**2.1.2.2 Instantaneous temperature distribution model of the ladle bottom** Heat transfer in the ladle bottom can be considered as a one-dimensional unsteady heat conduction [2, 3, 35], formulated as

$$\frac{\partial T_b}{\partial \tau} = \frac{1}{\rho_b c_b} \left[ \frac{1}{\partial z} \left( \lambda_b \cdot \frac{\partial T_b}{\partial z} \right) \right] \quad (7)$$

$$\text{when } z = 0, \quad T_b = T_{st} \quad (8)$$

$$\text{when } z = h_b, \quad \lambda_b \frac{\partial T_b}{\partial z} = -\alpha_{b-en}(T_b - T_\infty) \quad (9)$$

$$T_b(z, 0) = f(z) \quad (10)$$

Equation (7) is the heat conduction differential equation of the ladle bottom, where  $T_b$ ,  $\rho_b$ ,  $\lambda_b$ , and  $c_b$  are, respectively, the temperature in °C, density in kg/m<sup>3</sup>, heat conductivity in W/(m °C), and specific heat in J/(kg °C) of the ladle bottom. Equations (8) and (9) are the boundary conditions for Eq. (7), where  $h_b$  is the thickness of the ladle bottom in m; and  $\alpha_{b-en}$  is the convection heat transfer coefficient between the steel shell of the ladle bottom and the environment in W/(m<sup>2</sup> °C). Equation (10) is the initial condition. See the existing literature [2] for more details.

**2.1.2.3 Thermal loss from the ladle lining** Thermal loss from the ladle lining is obtained by calculating the heat flow of the contact surface between the ladle lining and the molten steel as

$$Q_{lin} = \lambda_{w,1} \frac{\partial T_w}{\partial r} \Big|_{r=r_1} 2\pi r_1 h + \lambda_{b,1} \frac{\partial T_b}{\partial z} \Big|_{z=0} \pi r_1^2 \quad (11)$$

where  $Q_{lin}$  is the thermal loss from the ladle lining in W;  $h$  is the height of the steel bath in m;  $\lambda_{w,1}$  is the heat conductivity of the ladle wall material that is in contact with the molten steel in W/(m °C); and  $\lambda_{b,1}$  is the heat conductivity of the ladle bottom material that is in contact with the molten steel in W/(m °C).

### 2.1.3 Thermal loss from the top surface

The top surface thermal loss mostly results from the radiation loss through the bare molten steel surface and slag surface. It is difficult to do an exact calculation for this loss using traditional mechanistic models. So, here the cooling water energy change is used to indirectly calculate this part of thermal loss as

$$Q_{sur} = \eta_{sur} c_{cw} F_{cw} \Delta T_{cw} \quad (12)$$

where  $Q_{sur}$  is the top surface thermal loss in W;  $\eta_{sur}$  is the correction coefficient;  $c_{cw}$ ,  $F_{cw}$ , and  $\Delta T_{cw}$  are the specific heat in J/(kg °C), flow rate in kg/s, and temperature difference in °C between inlet and outlet of the cooling water. Through analysis,  $\eta_{sur}$  can be considered as a function of three online available variables,  $F_{arg}$  (the argon flow rate in Nm<sup>3</sup>/s),  $D_{sl-co}$  (the distance between the steel bath and the ladle cover in m), and  $T_{st}$  as

$$\eta_{sur} = f_{sur}(F_{arg}, D_{sl-co}, T_{st}) \quad (13)$$

Similar with  $f_{arc}$ , the concrete expression of  $f_{sur}$  is also hard to derive by mechanistic approaches.

### 2.1.4 Thermal effects resulting from the additions

The additions include the slag and metal alloys. Their thermal effects are calculated as

$$Q_{add} = \left( \sum_i \frac{1}{\tau_i} m_i k_i \right) m_{st} c_{st} \quad (14)$$

where  $Q_{add}$  is the total thermal effect of additions in W;  $i$  denotes a specific addition with mass  $m_i$  in kg, and  $k_i$  is its temperature influence coefficient in °C/kg;  $\tau_i$  is the time that addition  $i$  takes to reach the steel bath temperature in s; and  $m_{st}$  and  $c_{st}$  are the mass in kg and specific heat in J/(kg K) of molten steel respectively. The value of  $k_i$  can be obtained by statistical analysis based on actual production data. Table 1 shows the temperature influence coefficients  $k_i$ s of various additions for the LF system considered in this study.

### 2.1.5 Thermal loss due to stirring-argon injection

The thermal loss due to stirring-argon injection is calculated as

$$Q_{arg} = c_{arg} F_{arg} (T_{st} - T_{arg}) \quad (15)$$

where  $Q_{arg}$  is the heat flow carried away by argon in W; and  $c_{arg}$  and  $T_{arg}$  are the specific heat in J/(Nm<sup>3</sup> °C) and initial temperature in °C of argon respectively.

### 2.1.6 The overall mechanistic thermal model

According to the energy conservation law, the following mechanistic thermal model can be obtained by combining all of the above factors

$$\frac{dT_{st}}{d\tau} = \frac{Q_{arc} - Q_{lin} - Q_{sur} - Q_{add} - Q_{arg}}{c_{st} m_{st} + c_{sl} m_{sl}} \quad (16)$$

where  $m_{sl}$  and  $c_{sl}$  are the mass in kg and specific heat in J/(kg K) of the slag respectively.

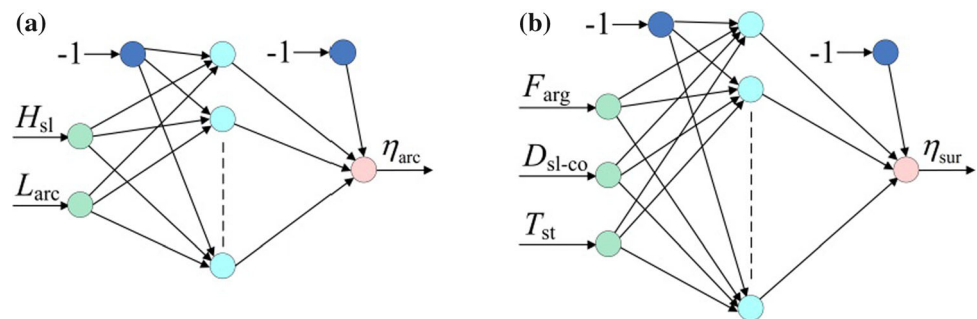
## 2.2 Development of empirical part using indirect training method

Obviously, the above mechanistic thermal model cannot be immediately used for predicting the temperature of molten steel, as there are two unknown functions, namely  $f_{arc}$  and  $f_{sur}$ . In this article, two single-hidden layer feed-forward neural networks (SLFNs) are utilized to respectively approximate  $f_{arc}$  and  $f_{sur}$  (see Fig. 2), for such neural networks can approximate any nonlinear relationships



**Table 1** Temperature influence coefficients of additions

Addition	$k_f s$ ( $10^{-2}$ °C/kg)	Addition	$k_f s$ ( $10^{-2}$ °C/kg)
High carbon ferrochrome	− 0.95	Low carbon ferrochromium	− 0.65
High carbon ferromanganese	− 0.9	Low carbon ferromanganese	− 0.75
Ferromolybdenum	− 0.75	Ferrosilicon	+ 0.9
Slag	− 1.0	Carbon powder	− 2.5
Al	+ 5.0	Ni	− 0.5
FeNb	− 0.35	FeTi	− 0.4
CaSi	− 1.05	FeAl	+ 1.0
HCZ1	− 1.0	Al–D	− 0.5
CaO	− 1.0	LCCR	− 0.65

**Fig. 2** Schematic representation of the two SLFN-based empirical models: **a**  $f_{\text{arc}}$ ; **b**  $f_{\text{sur}}$ 

arbitrarily well [36], and they have been successfully applied in modeling some LF metallurgy processes to predict the molten steel temperature [5, 6, 9, 10]. However, for practical LF metallurgy processes, only the values of the inputs to two empirical models,  $[H_{\text{sl}}, L_{\text{arc}}]$  and  $[F_{\text{arg}}, D_{\text{sl-co}}, T_{\text{st}}]$ , are available, whereas no target values of their outputs  $\eta_{\text{arc}}$  and  $\eta_{\text{sur}}$  are available. Therefore, traditional neural network training methods, which usually train the empirical part by directly minimizing the errors between the outputs of the empirical model(s) and its (their) target outputs, would be difficult to apply. In this study, the two SLFN-based empirical models are trained indirectly by minimizing the errors between the molten steel temperature predicted by the hybrid model and its measured values. The basic description of this indirect training method is given below.

The two SLFN-based empirical models of  $f_{\text{arc}}$  and  $f_{\text{sur}}$  can be formulated as

$$\eta_{\text{arc}} = \hat{f}_{\text{arc}}(H_{\text{sl}}, L_{\text{arc}}, \theta_{\text{arc}}) \quad (17)$$

$$\eta_{\text{sur}} = \hat{f}_{\text{sur}}(F_{\text{arg}}, D_{\text{sl-co}}, T_{\text{st}}, \theta_{\text{sur}}) \quad (18)$$

where  $\hat{f}_{\text{arc}}$  and  $\hat{f}_{\text{sur}}$  denote the SLFN-based empirical models used to approximate the unknown functions  $f_{\text{arc}}$  and  $f_{\text{sur}}$ , respectively, while  $\theta_{\text{arc}}$  and  $\theta_{\text{sur}}$  are the vectors of weights and bias of the corresponding SLFNs.

In essence, to train  $\hat{f}_{\text{arc}}$  and  $\hat{f}_{\text{sur}}$  is to determine the optimal values of their parameters  $\theta_{\text{arc}}$  and  $\theta_{\text{sur}}$ . The indirect method fulfills the training task as follows. Firstly, Eqs. (17) and (18) are, respectively, substituted into Eqs. (1) and (12), so that a hybrid prediction model is obtained

$$\frac{dT_{\text{st}}}{dt} = \frac{\hat{f}_{\text{arc}}(H_{\text{sl}}, L_{\text{arc}}, \theta_{\text{arc}})P_{\text{arc}} - Q_{\text{lin}} - \hat{f}_{\text{sur}}(F_{\text{arg}}, D_{\text{sl-co}}, T_{\text{st}}, \theta_{\text{sur}})c_{\text{cw}}F_{\text{cw}}\Delta T_{\text{cw}} - Q_{\text{add}} - Q_{\text{arg}}}{c_{\text{sl}}m_{\text{st}} + c_{\text{sl}}m_{\text{sl}}} \quad (19)$$

Then,  $\theta_{\text{arc}}$  and  $\theta_{\text{sur}}$  are regarded as the vectors of parameters to be identified in the hybrid prediction model, and further they are determined by using the proposed IICS algorithm to minimize an objective function which is defined with the measurements of the molten steel temperature, as shown below.

$$J(\theta_{\text{arc}}, \theta_{\text{sur}}) = \frac{\sum_{h=1}^H \sum_{m=1}^{M_h} |x_{hm} - \hat{x}_{hm}|}{\sum_{h=1}^H M_h} \quad (20)$$

where  $H$  is the number of heats of training data;  $M_h$  is the number of temperature samples of molten steel in the  $h$ th heat;  $x$  represents the measured value; and  $\hat{x}$  represents the hybrid model prediction. In such way, the training of  $\hat{f}_{\text{arc}}$  and  $\hat{f}_{\text{sur}}$  can be appropriately fulfilled while avoiding the requirement of their target outputs.

### 3 Information interaction-enhanced CS

In this section, a brief overview of basic CS (BCS) is first given for self-completeness. Next, the proposed information interaction-enhanced CS (IICS) algorithm is elaborated, followed by the complexity analysis of IICS.

#### 3.1 BCS algorithm

Cuckoo search (CS), introduced by Yang and Deb in 2009 [22], is one of the intelligent optimization algorithms. The core idea behind this algorithm is some cuckoo species' brood parasitism. Besides, the CS algorithm also incorporates into its framework the mathematical model of the Lévy flight behavior found in some birds and fruit flies. The following are some idealized rules adopted in CS development [22]: (1) the number of available host nests is fixed, and each cuckoo each time lays one egg in a randomly selected nest; (2) the nests containing high-quality eggs will be chosen to partake in the next generation; and (3) the egg laid by a cuckoo can be identified by the host bird with a probability  $p_a$ . Once it is identified, the host bird will either push the egg out or just discard the nest, and then make a new one. In addition, it is worthy to point out that a nest, an egg or a cuckoo is equivalent to a solution and only minimization problems are considered in the rest of the article without loss of generality.

Based on the rules above, Lévy flight is performed first to generate the new solution  $z_i^{\text{new}}$  for cuckoo  $i$  with the following formula:

$$z_i^{\text{new}} = z_i + \alpha \otimes \text{Levy}(\lambda) \quad (21)$$

where  $\alpha$  is a vector of step size scaling factors that should be related to the scales of the problem under consideration, and it can be in most cases used as [37–39]

$$\alpha = \alpha_0 (z_i - z_{\text{best}}) \quad (22)$$

where  $\alpha_0$  is a constant usually set as 0.01 [38, 39] and  $z_{\text{best}}$  represents the current best solution. In Eq. (21),  $\text{Levy}(\lambda)$  is a random vector drawn from a Lévy distribution,  $\lambda$  is a Lévy flight parameter, and  $\otimes$  represents the entry-wise multiplication.

In essence, Lévy flights offer a random walk with random steps drawn from a Lévy distribution. From the perspective of implementation, there are two procedures to generate random numbers using Lévy flights [38], that is, the selection of a random direction and the generation of steps that obey the selected Lévy distribution. The generation of a direction should be drawn from a uniform distribution, whereas the generation of steps is quite tricky. There are several ways of accomplishing this, but one of the most efficient and yet straightforward ways is to utilize

the Mantegna algorithm [40], in which the step size  $s$  is given by

$$s = \frac{u}{|v|^{1/\beta}} \quad (23)$$

where  $u$  and  $v$  are drawn from the following normal distributions:

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \quad (24)$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \cdot \beta \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta} \sigma_v = 1 \quad (25)$$

where  $\Gamma$  represents the Gamma function, and  $\beta$  is a distribution parameter related with  $\lambda$  in Eq. (21) as  $\lambda = 1 + \beta$  ( $0 < \beta \leq 2$ , and  $\beta = 1.5$  [38] in CS).

As per the above, Eq. (21) can be rewritten as

$$z_i^{\text{new}} = z_i + 0.01 \cdot s \cdot (z_i - z_{\text{best}}) \otimes r \quad (26)$$

where  $r$  is a random vector with all its elements generated from the standard normal distribution  $N(0, 1)$ , and  $s$  is calculated using Eq. (23).

After comparing the fitness values between each old and new solution at the same nest and retaining the solutions with lower fitness values, the new solution  $z_i^{\text{new}}$  for cuckoo  $i$  is generated again by imitating the action of alien egg discovery, which can be formulated as:

$$z_i^{\text{new}} = z_i + \text{rnd}_1 \cdot H(p_a - \text{rnd}_2) \otimes (z_j - z_k) \quad (27)$$

where  $z_j$  and  $z_k$  are two randomly selected solutions;  $p_a$  is a vector with all its elements being  $p_a$ ;  $\text{rnd}_1$  is a random number generated from the standard uniform distribution  $U(0, 1)$ ;  $\text{rnd}_2$  is a random vector with all elements generated from  $U(0, 1)$ ; and  $H(\bullet)$  is a step function, defined as

$$H(x) = H(x_1, x_2, \dots, x_d, \dots, x_D) \\ = [H_1, H_2, \dots, H_d, \dots, H_D] \quad (28)$$

where  $H_d = 1$  if  $x_d > 0$ , otherwise  $H_d = 0$ . More details on BCS can be found in [38].

#### 3.2 IICS algorithm

As mentioned above, information interaction between individuals is lacking in the search process of BCS, while it is well known that the information interaction between people plays an important role for their team to accomplish an assignment with high efficiency. Accordingly, it is expected that a CS with an information interaction-enhanced mechanism can realize a better search performance than BCS. Based on this consideration, a modified CS called information interaction-enhanced CS (IICS) is proposed in this article. In IICS, cuckoo  $i$  is offered an opportunity to get some potentially useful information from a selected information provider  $z_{pi}$  (here  $p_i$  defines

which cuckoo should be selected as the information provider for cuckoo  $i$ ). To get the  $z_{pi}$  for cuckoo  $i$ , three candidates are chosen from  $z = [z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_{N_p}]$  randomly, where  $N_p$  denotes the population size. Among these three candidates, the one with the best (i.e., smallest) fitness value is selected as  $z_{pi}$ , and meanwhile its index is assigned to  $p_i$ .

The main differences between IICS and BCS are the updating formulas of the Lévy flight and alien egg discovery. To be specific, in IICS, the updating formulas used in BCS, namely Eqs. (26) and (27), are replaced with Eqs. (29) and (30), respectively.

$$z_i^{\text{new}} = z_i + \Delta z_i^L \otimes (I - H(c_a - \text{rnd}_3)) + \Delta z_i^{I_1} \otimes H(c_a - \text{rnd}_3) \quad (29)$$

$$z_i^{\text{new}} = z_i + H(p_a - \text{rnd}_2) \otimes (\Delta z_i^A \otimes (I - H(c_a - \text{rnd}_4)) + \Delta z_i^{I_2} \otimes H(c_a - \text{rnd}_4)) \quad (30)$$

where

$$\Delta z_i^L = 0.01 \cdot s \cdot (z_i - z_{\text{best}}) \otimes r \quad (31)$$

$$\Delta z_i^{I_1} = \text{rnd}_5 \otimes (z_{p_i} - z_i) \quad (32)$$

$$\Delta z_i^A = \text{rnd}_1 \cdot (z_j - z_k) \quad (33)$$

$$\Delta z_i^{I_2} = \text{rnd}_6 \otimes (z_{p_i} - z_i) \quad (34)$$

In the above equations,  $I$  is a vector with each element being one;  $c_a$  is a vector with all elements being  $c_a$ , which is a coefficient for adjusting the combination of the proposed information interaction-enhanced mechanism and Lévy flights (or the alien egg discovery);  $\text{rnd}_3$ ,  $\text{rnd}_4$ ,  $\text{rnd}_5$ , and  $\text{rnd}_6$  are random vectors with all their elements drawn from  $U(0, 1)$ ; and  $H(\bullet)$  is the same step function as defined in Eq. (28).

Figure 3 presents the complete implementation of the proposed IICS. Firstly, various parameters are set and a group of  $N_p$  solutions are randomly initialized (lines 1–2). Next, the fitness function value for each solution is calculated, and the one with the best fitness in the current population is assigned to  $z_{\text{best}}$  (lines 3–4). Thereafter, these solutions are sequentially evolved with the first (lines 7–9) and second (line 14) proposed search operators, which respectively combine the information interaction-enhanced mechanism with Lévy flights and the alien egg discovery. Following the generation of each new solution, the optimal selection between the old and newly-generated solutions at the same nest will be performed, and the one with the smaller fitness value is retained (lines 10–13 and 15–18). Finally,  $z_{\text{best}}$  is updated and it is selected as the optimal solution of the search process (lines 19–21).

#### Algorithm 1. Implementation of IICS.

```

1: Set the population size  $N_p$ , maximum iteration number  $g_{\max}$ ,
   discovering probability  $p_a$ , adjustment coefficient  $c_a$ , search
   space dimension  $D$ ;
2: Initialize a population of  $N_p$  solutions  $z_i$  ( $i = 1, 2, \dots, N_p$ );
3: Calculate the fitness value of each solution  $f_i$  ( $i = 1, 2, \dots, N_p$ );
4: Find out the current best solution, and let  $z_{\text{best}}$  and  $f_{\min}$  be
   equal to this solution and its fitness value, respectively;
5:  $g = 0$ ;
6: while  $g < g_{\max}$  do
7:   for  $i = 1$  to  $N_p$  do
8:     Select an information provider  $z_{pi}$ ;
9:     Generate a new solution  $z_i^{\text{new}}$  using Eq. (29);
10:    Calculate its fitness value  $f_i^{\text{new}}$ ;
11:    if  $f_i^{\text{new}} < f_i$  then
12:       $z_i = z_i^{\text{new}}$  and  $f_i = f_i^{\text{new}}$ ;
13:    end if
14:    Generate a new solution  $z_i^{\text{new}}$  using Eq. (30);
15:    Calculate its fitness value  $f_i^{\text{new}}$ ;
16:    if  $f_i^{\text{new}} < f_i$  then
17:       $z_i = z_i^{\text{new}}$  and  $f_i = f_i^{\text{new}}$ ;
18:    end if
19:    if  $f_i < f_{\min}$  then
20:       $z_{\text{best}} = z_i$  and  $f_{\min} = f_i$ ;
21:    end if
22:  end for
23:   $g = g + 1$ ;
24: end while

```

Fig. 3 Complete framework of the proposed IICS algorithm

### 3.3 Complexity analysis of IICS

To facilitate the analysis, the computation time complexity of BCS is given firstly in this section. For each cuckoo,  $O(D)$  number of operations are performed in an iteration in BCS, resulting in  $O(N_p, D)$  complexity. However, generally CS runs for a number of iterations, so the overall complexity depends on the maximum iteration number ( $g_{\max}$ ). This procedure gives the overall time complexity of BCS as  $O(N_p, D, g_{\max})$ . Compared with BCS, our IICS needs to perform additional computations of  $O(N_p, D, g_{\max})$  for the proposed information interaction-enhanced mechanism. Meanwhile, the selection of information providers consumes further computational complexity of  $O(N_p, D, g_{\max})$ . Accordingly, the computation time complexity of IICS is the same as that of BCS, i.e.,  $O(N_p, D, g_{\max})$ . However, IICS significantly outperforms BCS according to the experimental results given in the following



sections. These observations suggest that when compared with BCS, our IICS achieves better tradeoff between performance improvement and computation time complexity.

#### 4 Optimizing parameters of the hybrid model using IICS

In this section, the IICS algorithm is applied to solve the parameter optimization problem of empirical models  $\hat{f}_{arc}$  and  $\hat{f}_{sur}$  in the hybrid prediction model described by Eq. (19), hereafter referred as to the PO problem, and thus accomplish their training with the proposed indirect method. In the process of search, nest  $i$  is encoded as  $z_i = [\theta_{arci}, \theta_{suri}]$  to represent a set of candidate values of  $\theta_{arc}$  and  $\theta_{sur}$ , and Eq. (20) is used as the fitness function to evaluate the quality of each nest. Algorithm 2, as illustrated in Fig. 4, describes the procedures for the calculation of the fitness value  $J_i$  for nest  $i$ .

The implementation of IICS for optimizing the parameters in the empirical part with the indirect training method is given by Algorithm 3, as illustrated in Fig. 5.

#### 5 Experiments

In this section, the IICS algorithm was validated on 16 classical benchmark functions and the 29 CEC 2017 benchmark functions. Then, the actual production data

##### Algorithm 2. Calculation for $J_i$ .

**Input:**  $z_i = [\theta_{arci}, \theta_{suri}]$ .

**Output:**  $J_i$ .

- 1: Substitute the values of the parameter vectors  $\theta_{arc}$  and  $\theta_{sur}$  corresponding to nest  $i$  (i.e.,  $\theta_{arci}$  and  $\theta_{suri}$ ) into the hybrid prediction model described by Eq. (19);
- 2: **for**  $h = 1$  to  $H$  **do**
- 3: Based on the initial offline measurement of the molten steel temperature as well as the online measurements of other required process variables, alternately solve the hybrid prediction model using the Euler method and the unsteady heat conduction equations of the ladle wall and bottom using the finite difference method explicit scheme, so as to obtain the predicted values of the  $h$ th heat of molten steel temperature at each offline sampling point, i.e.,  $\hat{x}_{h1}, \hat{x}_{h2}, \dots, \hat{x}_{hM_h}$ ;
- 4: **end for**
- 5:  $J_i = \sum_{h=1}^H \sum_{m=1}^{M_h} |x_{hm} - \hat{x}_{hm}| / \sum_{h=1}^H M_h$ .

Fig. 4 Calculation procedures for the  $i$ th nest's fitness value

##### Algorithm 3. Implementation of IICS for PO.

- 1: Set the population size  $Np$ , maximum iteration number  $g_{max}$ , discovering probability  $p_a$ , adjustment coefficient  $c_a$ , and number of model parameters that need to be estimated  $D$ ;
- 2: Initialize a population of  $Np$  host nests  $z_i = [\theta_{arci}, \theta_{suri}]$  ( $i = 1, 2, \dots, Np$ );
- 3: **for**  $i = 1$  to  $Np$  **do**
- 4: Calculate  $J_i$  using Algorithm 2;
- 5: **end for**
- 6: Find out the current best solution, and let  $z_{best}$  and  $J_{min}$  be equal to this solution and its fitness value, respectively;
- 7:  $g = 0$ ;
- 8: **while**  $g < g_{max}$  **do**
- 9: **for**  $i = 1$  to  $Np$  **do**
- 10: Select an information provider  $z_{pi}$ ;
- 11: Generate a new solution  $z_i^{new}$  using Eq. (29);
- 12: Calculate its fitness value  $J_i^{new}$  using Algorithm 2;
- 13: **if**  $J_i^{new} < J_i$  **then**
- 14:  $z_i = z_i^{new}$  and  $J_i = J_i^{new}$ ;
- 15: **end if**
- 16: Generate a new solution  $z_i^{new}$  using Eq. (30);
- 17: Calculate its fitness value  $J_i^{new}$  using Algorithm 2;
- 18: **if**  $J_i^{new} < J_i$  **then**
- 19:  $z_i = z_i^{new}$  and  $J_i = J_i^{new}$ ;
- 20: **end if**
- 21: **if**  $J_i < J_{min}$  **then**
- 22:  $z_{best} = z_i$  and  $J_{min} = J_i$ ;
- 23: **end if**
- 24: **end for**
- 25:  $g = g + 1$ ;
- 26: **end while**
- 27:  $[\theta_{arc}, \theta_{sur}] = z_{best}$ .

Fig. 5 Implementation of the IICS algorithm for solving the PO problem

from a 300 t LF at Baoshan Iron & Steel Co. Ltd were used to build the proposed hybrid prediction model using IICS.

#### 5.1 Performance validation of IICS on benchmark functions

##### 5.1.1 Validation on classical benchmark functions

In this subsection, 16 classical benchmark functions [41–43] are employed for investigating the performance of IICS. These benchmark functions are listed in Table 2, which includes the mathematical formula, search range, and function value at the global minimum ( $F^*$ ) of each benchmark function. These benchmark functions fall into

**Table 2** Classical benchmark functions used

Function name	Mathematical formula	Range	$F^*$
Sphere function	$f_1(\mathbf{x}) = \sum_{d=1}^D x_d^2$	$[-100, 100]^D$	0
Schwefel's problem 2.22	$f_2(\mathbf{x}) = \sum_{d=1}^D  x_d  + \prod_{d=1}^D  x_d $	$[-10, 10]^D$	0
High conditioned elliptic function	$f_3(\mathbf{x}) = \sum_{d=1}^D (10^6)^{\frac{d-1}{D-1}} x_d^2$	$[-100, 100]^D$	0
Schwefel's problem 1.2	$f_4(\mathbf{x}) = \sum_{d=1}^D (\sum_{j=1}^d x_j)^2$	$[-100, 100]^D$	0
Sum square function	$f_5(\mathbf{x}) = \sum_{d=1}^D dx_d^2$	$[-10, 10]^D$	0
Griewank's function	$f_6(\mathbf{x}) = \sum_{d=1}^D \frac{x_d^2}{4000} - \prod_{d=1}^D \cos(\frac{x_d}{\sqrt{d}}) + 1$	$[-600, 600]^D$	0
Rastrigin's function	$f_7(\mathbf{x}) = \sum_{d=1}^D (x_d^2 - 10 \cos(2\pi x_d) + 10)$	$[-5.12, 5.12]^D$	0
Noncontinuous Rastrigin's function	$f_8(\mathbf{x}) = \sum_{d=1}^D (y_d^2 - 10 \cos(2\pi y_d) + 10), \quad y_d = \begin{cases} x_d &  x_d  < \frac{1}{2} \\ \frac{\text{round}(2x_d)}{2} &  x_d  \geq \frac{1}{2} \end{cases}$	$[-5.12, 5.12]^D$	0
Ackley's function	$f_9(\mathbf{x}) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{d=1}^D x_d^2}) - \exp(\frac{1}{D} \sum_{d=1}^D \cos(2\pi x_d)) + 20 + e$	$[-32.768, 32.768]^D$	0
Schwefel's function	$f_{10}(\mathbf{x}) = 418.9828872724336 \cdot D - \sum_{d=1}^D x_d \sin( x_d ^{\frac{1}{2}})$	$[-500, 500]^D$	0
Weierstrass function	$f_{11}(\mathbf{x}) = \sum_{d=1}^D (\sum_{k=0}^{k_{\max}} a^k \cos(2\pi b^k (x_d + 0.5))) - D \sum_{k=0}^{k_{\max}} a^k \cos(2\pi b^k \cdot 0.5)$ where $a = 0.5, b = 3, k_{\max} = 20$	$[-0.5, 0.5]^D$	0
Generalized penalized function1	$f_{12}(\mathbf{x}) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{d=1}^{D-1} (y_d - 1)^2 \cdot [1 + 10 \sin^2(\pi y_{d+1})] + (y_D - 1)^2 \right\}$ $+ \sum_{d=1}^D u(x_d, 10, 100, 4)$ where $y_d = 1 + \frac{1}{4}(x_d + 1), u(x_d, a, k, m) = \begin{cases} k(x_d - a)^m & x_d > a \\ 0 & -a \leq x_d \leq a \\ k(-x_d - a)^m & x_d < -a \end{cases}$	$[-50, 50]^D$	0
Generalized penalized function2	$f_{13}(\mathbf{x}) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{d=1}^{D-1} (x_d - 1)^2 \cdot [1 + \sin^2(\pi x_{d+1})] + \dots \cdot (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \} + \sum_{d=1}^D u(x_d, 5, 100, 4)$	$[-50, 50]^D$	0
Schaffer function	$f_{14}(\mathbf{x}) = 0.5 + (\sin^2(\sqrt{\sum_{d=1}^D x_d^2}) - 0.5) / (1 + 0.001(\sum_{d=1}^D x_d^2))^2$	$[-100, 100]^D$	0
Alpine function	$f_{15}(\mathbf{x}) = \sum_{d=1}^D  x_d \cdot \sin(x_d) + 0.1 \cdot x_d $	$[-10, 10]^D$	0
Levy function	$f_{16}(\mathbf{x}) = \sum_{d=1}^{D-1} (x_d - 1)^2 \cdot [1 + \sin^2(3\pi x_{d+1})] + \sin^2(3\pi x_1) +  x_D - 1  [1 + \sin^2(3\pi x_D)]$	$[-10, 10]^D$	0

two categories: unimodal problems and multimodal problems. Of them,  $f_1, f_2, f_3, f_4$ , and  $f_5$  are unimodal ones containing only one optimum, whereas the rest 11 benchmark functions, namely  $f_6$  to  $f_{16}$ , are multimodal ones having many local optima, but only one global optimum. In addition, it should be pointed out that the validation is performed with 50 variables, that is, the dimension ( $D$ ) of each benchmark function is 50.

In order to show the competitiveness of the proposed IICS, the proposed IICS is compared with six recently developed CS variants, i.e., CCS [43], ACS [44], ICS [45], NNCS [46], BHCS [47], and HECS [48], as well as BCS in this subsection. For fair comparison,  $Np$  and  $g_{\max}$  (the maximum iteration number used to terminate the iteration) of the eight CS-based algorithms are set to be same, that is  $Np = D$  and  $g_{\max} = 5000$  as done in the studies [46, 49]. We follow the parameter settings of CCS, ACS, ICS, NNCS, BHCS, and HECS used or recommended in the studies conducted on them. For BCS and IICS, the configuration of the common parameter  $p_a$  is taken from that originally utilized by Yang and Deb [22, 38]. From these studies,  $p_a = 0.25$  is a better choice for most optimization problems. The specific parameter of IICS, i.e.,  $c_a$ , is adjusted via experiment analysis. According to the experiments, it was found that when  $c_a = 0.08$  the search performance of IICS can be well balanced on different kinds of optimization problems. The parameter settings of the eight CS-based algorithms are summarized in Table 3.

To reduce random discrepancy, each algorithm is performed independently for 30 runs on each benchmark function, and the mean, best, worst, and standard deviation (SD) of the function error ( $f(\mathbf{x}_{\text{best}}) - F^*$ ), where  $\mathbf{x}_{\text{best}}$  represents the best solution achieved by the algorithm in a run, are calculated and recorded. The test results of the eight CS-based algorithms on the 16 classical benchmark functions are shown in Table 4. The best mean error values among the eight algorithms are highlighted in boldface. On these 16 classical benchmark functions, IICS produced the lowest mean error values for eight of these, while BHCS produces the lowest mean error values in five cases. For two of the benchmark functions, IICS and some of its competitors are seen to produce equal results. For  $f_{14}$ , it is observed that CCS produces the better solution. Although BHCS attains the best mean results on 4 out of the 5 unimodal benchmark functions, it sacrifices performance on multimodal ones. On the contrary, IICS is not only very efficient in solving the unimodal problems but also attain very competitive performance on the multimodal ones. To statistically compare IICS with each competitor, the multiple-problem Wilcoxon rank test is conducted at a significance level of 5% based on the mean error value. In Tables 4 and 5, the statistical significance state is indicated with the symbols +,  $\approx$ , and –, denoting that the

competitor performs significantly worse than, insignificantly different from, and significantly better than the IICS algorithm respectively. Moreover, the Friedman test is also applied to determine the differences between these algorithms and rank them with a significance level of 0.05. Table 4 indicates that IICS provides significantly better results than all of the other seven algorithms. The row headed “Mean Rank” provides the final ranking of different algorithms for all 16 classical benchmark functions. The results show that IICS, NNCS and BHCS are in the first, second and third orders with 1.56, 3.84, and 3.91 mean rank values respectively. In addition, the  $p$  value ( $7.4999\text{e} - 09$ ) is smaller than the chosen significance level (0.05). This indicates that there is at least one significant difference among the algorithms’ results.

To show the convergence process visually, the convergence curves of each algorithm in term of the mean error values on the classical benchmark functions are presented in Fig. 6. For the ease of comparison, semilogarithmic coordinate is used to plot the convergence curves of each benchmark function, except that of  $f_{14}$ . When the convergence curves in Fig. 6 are analyzed, it can be observed that IICS performs well in 14 out of the 16 classical benchmark functions. IICS has significantly higher convergence speeds in  $f_6, f_7, f_9$  to  $f_{13}$ , and  $f_{16}$  compared with the other seven CS-based algorithms. Therefore, IICS can be regarded efficient. However, the result patterns slightly differ in some of the 14 functions: for  $f_{14}$ , it can be seen that the rapid convergence of IICS is at the expense of being trapped in the local minima, while the results of  $f_1$  to  $f_3, f_5$ , and  $f_8$  show that IICS and BHCS perform similarly in terms of convergence speed and accuracy. In addition, it can also be observed that IICS is weak in  $f_4$  and  $f_{15}$ . For these two functions, BHCS has the highest convergence speeds and best search results.

### 5.1.2 Validation on CEC 2017 benchmark functions

In this subsection, benchmark functions from CEC 2017 are used as the benchmark test set. This test set consists of 29 benchmark functions: two unimodal functions,  $F_1$  and  $F_3$ ; seven simple multimodal functions,  $F_4 - F_{10}$ ; ten hybrid functions,  $F_{11} - F_{20}$ ; and ten composition functions,  $F_{21} - F_{30}$ . The order of these functions is the same as the original article [50]. The detailed functions are not presented here to save space. These benchmark functions are considered difficult to optimize, as all of them are shifted and rotated, and some of them are hybrid or composition functions. It should be noted that the function  $F_2$  named “Shifted and Rotated Sum of Different Power” is not used here due to unstable behavior especially for higher dimensions, as described in [50]. The detailed definitions can be found in the original article.

**Table 3** Parameter settings of involved algorithms

Algorithm	Parameter settings
IICS	$Np = D, p_a = 0.25, \alpha = 0.01, c_a = 0.08$
BCS	$Np = D, p_a = 0.25, \alpha = 0.01$
CCS	$Np = D, p_a = 0.25$ , generation of chaotic sequences: Gauss map
ACS	$Np = D, p_a = 0.25$
ICS	$Np = D, p_{amax} = 0.5, p_{amin} = 0.005, \alpha_{max} = 0.5, \alpha_{min} = 0.01$
NNCS	$Np = D, p_a = 0.25, p = 0.25$ , selection of nearest neighbour solutions: fitness-based similar metrics
BHCS	$Np = D, p_a = 0.3, \alpha = 1.1, \beta = 1.7, \delta = 1.6, E = 1$
HECS	$Np = D, p_0 = 0.6, \alpha = 0.4, \beta = 1.5$ , proportion of exploitation group: 70%, proportion of exploration group: 30%
HFPSO	$Np = D, c_1 = c_2 = 1.49445, \omega_{max} = 0.9, \omega_{min} = 0.5, V_{max} = 0.1 * (X_{max} - X_{min}), V_{min} = -V_{max}$
ELSHADE-SPACMA	$Np^{init} = 18 * D, H = 5, Arc\_rate = 1.4, F_{CP} = 0.5, c = 0.8, p = 0.1, p^{init} = 0.3, p^{min} = 0.15$
HSES	CMA-ES: $Np = \lfloor 3 \ln D \rfloor + 80, N = Np/2, c_m = 1, \alpha_{cov} = 2$ univariate sampling: $Np = 200, N = 100, cc = 0.96, I = 20$ for the first step, and $Np = 450, N = 360$ for the fourth step
ISCA	$Np = D, A = 2 - 2 * (t/g_{max}), r_1 \in [0, 2\pi], C \in [0, 2], CR = 0.3$

In addition to BCS, CCS, ACS, ICS, NNCS, BHCS, and HECS, this subsection compares IICS with four other intelligent algorithms, i.e., hybrid firefly and particle swarm optimization (HFPSO) [51], enhanced LSHADE-SPACMA (ELSHADE-SPACMA) [52], hybrid sampling evolution strategy (HSES) [53], and improved sine cosine algorithm with crossover scheme (ISCA) [54], which belong to PSO, DE, covariance matrix adaptation evolution strategy (CMA-ES), and sine cosine algorithm (SCA) communities, respectively. These four algorithms have all proved their good performance on CEC 2017 benchmark functions, and HSES and ELSHADE-SPACMA have won the first and third places in the CEC 2018 competition respectively. Parameter configurations of the new selected algorithms are the same as in the corresponding references, as listed in Table 3. Size of the population ( $Np$ ) is set equal to the benchmark function dimension ( $D$ ) in IICS, BCS, CCS, ACS, ICS, NNCS, BHCS, HECS, HFPSO, and ISCA, while the settings of  $Np$  in ELSHADE-SPACMA and HSES are consistent with those in the original studies. In accordance with the original article of the competition of CEC 2017 problems [50], each algorithm is repeated 51 runs with the maximum number of function evaluations set to  $10,000 \times D$ .

Table 5 lists the obtained results from all involved algorithms on CEC 2017 benchmark functions with  $D = 50$ , including the mean, best, worst and standard deviation ( $SD$ ) values of the function error of every benchmark function obtained by each algorithm, as well as the findings from the multiple-problem Wilcoxon rank test and Friedman test both at a significance level of 0.05. In Table 5, the mean, best, worst and  $SD$  values for ELSHADE-SPACMA and HSES are collected from the

original articles [52, 53], and values smaller than  $10^{-8}$  are indicated as 0.00e+00. When the results from IICS compared with those from the other seven CS-based algorithms, it can be seen IICS produces the best mean error values for 17 of the 29 benchmark functions, while all the other seven CS-based competitors do so for just 13 benchmark functions. Examination of Symbol row in Table 5 further indicates that IICS achieves significantly better results than the other seven CS-based competitors, as well as HFPSO and ISCA. However, the ELSHADE-SPACMA and HSES algorithms are seen to return superior results to IICS. The findings from the Friedman test show that ELSHADE-SPACMA, HSES and IICS have the first, second, and third mean rank values of 1.69, 1.95, and 4.02, respectively.

## 5.2 Experimental verification based on actual production data

In this section, 537 heats of actual production data from a 300 t LF built in Baoshan Iron & Steel Co., Ltd., are employed to verify the ability of the proposed hybrid prediction model, as well as the performance of the IICS algorithm. Among these data, 437 heats are randomly selected for the development of the proposed prediction model, and the remainders are utilized for testing its performance.

The parameter setting for IICS to solve the PO problem is as follows:  $p_a = 0.25, c_a = 0.08, N_p = D$  (where  $D = [(2 + 1) \times hd_1 + (hd_1 + 1) \times 1] + [(3 + 1) \times hd_2 + (hd_2 + 1) \times 1]$ , and  $hd_1$  and  $hd_2$  denote the hidden neuron numbers of the empirical models  $\hat{f}_{arc}$  and  $\hat{f}_{sur}$  respectively), and  $g_{max} = 5000$ . Moreover, it should be

**Table 4** Comparison among eight CS-based algorithms on 16 classical benchmark functions

		IICS	BCS	CCS	ACS	ICS	NNCS	BHCS	HECS
$f_1$	Mean	9.1045e−80	1.2103e−16	5.2092e−27	3.4943e−15	1.1492e−26	5.7032e−26	<b>7.6718e−87</b>	9.0816e−13
	Best	6.4047e−82	7.9400e−17	1.4799e−27	1.5652e−19	4.5444e−27	2.5441e−26	8.3763e−99	4.0290e−15
	Worst	1.0474e−78	1.8044e−16	1.2553e−26	6.6248e−14	2.4916e−26	1.3247e−25	2.2727e−85	7.2246e−12
	SD	1.9806e−79	2.7493e−17	2.3640e−27	1.2026e−14	5.0936e−27	2.4366e−26	4.0779e−86	1.6631e−12
$f_2$	Mean	<b>1.0197e−48</b>	1.0391e−07	3.9451e−17	1.9589e−07	4.6875e−15	2.2108e−15	7.7583e−42	9.6716e−12
	Best	2.3713e−49	7.0947e−08	1.9001e−17	1.1735e−09	1.9344e−15	1.2576e−15	5.9656e−51	3.4226e−15
	Worst	3.3463e−48	1.4856e−07	7.0677e−17	8.5147e−07	7.3031e−15	3.1005e−15	2.1029e−40	1.3383e−10
	SD	7.2959e−49	1.9099e−08	1.3230e−17	2.0975e−07	1.2799e−15	5.3815e−16	3.7680e−41	2.8826e−11
$f_3$	Mean	5.2762e−76	2.1603e−13	2.4923e−24	2.2299e−09	3.4298e−24	5.0435e−23	<b>1.3185e−81</b>	7.5079e−09
	Best	3.0703e−78	1.2269e−13	7.3070e−25	4.8189e−15	9.9390e−25	1.9611e−23	4.1591e−96	6.4759e−12
	Worst	8.6207e−75	3.7299e−13	5.4977e−24	5.5524e−08	1.0259e−23	1.1006e−22	3.9357e−80	1.4279e−07
	SD	1.5497e−75	5.7190e−14	1.2663e−24	9.9728e−09	1.6792e−24	2.2959e−23	7.0636e−81	2.5881e−08
$f_4$	Mean	5.5204e+02	1.5243e+03	7.6669e+03	5.6889e+03	2.9010e+03	5.5265e+03	<b>1.7722e−03</b>	5.4455e+00
	Best	2.8713e+02	1.2438e+03	5.2771e+03	4.0088e+03	2.3548e+03	3.8692e+03	5.4896e−10	1.2992e+00
	Worst	8.9263e+02	1.7735e+03	1.0320e+04	7.8461e+03	3.4970e+03	7.3163e+03	4.5628e−02	3.8112e+01
	SD	1.2193e+02	1.3384e+02	1.2547e+03	1.0425e+03	3.2392e+02	7.3979e+02	8.2510e−03	6.8928e+00
$f_5$	Mean	1.7782e−80	2.2331e−17	9.2233e−28	1.7569e−15	1.8771e−27	1.0736e−26	<b>4.7950e−87</b>	2.7826e−13
	Best	4.4669e−82	1.2664e−17	3.1723e−28	2.6128e−20	9.3270e−28	3.5558e−27	1.8587e−98	7.4386e−16
	Worst	2.1195e−79	3.3418e−17	1.4909e−27	1.5123e−14	3.4685e−27	3.3911e−26	1.1433e−85	2.5486e−12
	SD	4.0074e−80	5.4560e−18	3.3132e−28	3.8722e−15	7.1922e−28	5.5310e−27	2.0657e−86	5.5680e−13
$f_6$	Mean	<b>0</b>	2.3054e−11	<b>0</b>	8.4484e−09	<b>0</b>	<b>0</b>	4.6835e−02	9.5708e−02
	Best	0	1.4333e−13	0	3.8200e−11	0	0	0	2.1316e−14
	Worst	0	2.4082e−10	0	1.0363e−07	0	0	1.7450e−01	7.0113e−01
	SD	0	4.7534e−11	0	2.2249e−08	0	0	5.0735e−02	1.4001e−01
$f_7$	Mean	<b>0</b>	9.2136e+01	1.0841e+02	1.3883e+02	4.2197e+01	6.0092e+01	1.6152e+01	2.4846e+01
	Best	0	7.1040e+01	9.1742e+01	1.1633e+02	3.5633e+01	4.3982e+01	8.9546e+00	1.6914e+01
	Worst	0	1.1068e+02	1.2691e+02	1.5305e+02	4.9667e+01	7.2254e+01	2.3879e+01	3.8803e+01
	SD	0	9.7487e+00	1.0097e+01	9.7594e+00	3.4288e+00	7.4029e+00	4.0593e+00	5.8489e+00
$f_8$	Mean	<b>0</b>	7.8532e+01	9.2884e+01	1.2276e+02	4.2404e+01	5.0187e+01	<b>0</b>	3.1447e+01
	Best	0	6.2334e+01	7.5442e+01	1.0393e+02	3.6342e+01	3.4660e+01	0	2.0000e+01
	Worst	0	8.9813e+01	1.1103e+02	1.3969e+02	4.7325e+01	6.0928e+01	0	6.0270e+01
	SD	0	6.3229e+00	1.0027e+01	1.0535e+01	2.6840e+00	6.5996e+00	0	8.1190e+00
$f_9$	Mean	<b>5.4475e−15</b>	3.1977e−03	4.7133e−14	5.7452e−06	1.1227e−13	1.9765e−13	3.7426e+00	5.0417e+00
	Best	3.5527e−15	3.6544e−04	3.1974e−14	1.9448e−06	7.8160e−14	1.2079e−13	2.7231e+00	3.0665e+00
	Worst	7.1054e−15	1.9237e−02	6.7502e−14	1.1932e−05	1.5632e−13	3.6948e−13	5.5088e+00	7.6055e+00
	SD	1.7724e−15	3.4682e−03	7.2190e−15	2.3901e−06	1.9079e−14	5.3200e−14	7.4785e−01	1.0869e+00
$f_{10}$	Mean	<b>4.0651e−09</b>	4.8286e+03	5.4499e+03	5.7974e+03	3.5282e+03	1.4048e+03	4.5595e+03	2.0753e+03
	Best	7.2760e−12	3.6889e+03	4.8528e+03	5.1737e+03	3.1088e+03	1.2967e+02	3.2176e+03	1.1844e+03
	Worst	7.0719e−08	5.3482e+03	6.2488e+03	6.5433e+03	3.8670e+03	2.7235e+03	7.4403e+03	3.0824e+03
	SD	1.5320e−08	3.1425e+02	3.3846e+02	3.1304e+02	2.1750e+02	6.1043e+02	7.4167e+02	4.9162e+02
$f_{11}$	Mean	<b>0</b>	7.4298e−01	7.4711e−12	7.1490e−03	6.2282e−05	4.5858e−12	6.1840e+00	4.3903e+00
	Best	0	4.7476e−01	8.1002e−13	2.1657e−03	2.8448e−05	6.6791e−13	2.7885e+00	2.2968e+00
	Worst	0	1.0156e+00	5.9885e−11	1.5243e−02	1.2768e−04	2.1075e−11	1.0476e+01	7.0250e+00
	SD	0	1.3643e−01	1.1266e−11	2.5551e−03	2.1668e−05	5.1091e−12	1.5537e+00	1.2290e+00
$f_{12}$	Mean	<b>0</b>	3.0325e+01	8.8873e−02	2.5756e+01	9.5626e−01	1.3380e−13	1.6400e−01	3.3385e−01
	Best	0	1.5435e+01	1.0239e−09	1.6084e+01	2.5272e−01	1.6678e−18	0	3.9151e−16
	Worst	0	4.4944e+01	8.0876e−01	4.0014e+01	2.3725e+00	1.5965e−12	1.4974e+00	2.3018e+00
	SD	0	7.5398e+00	2.1442e−01	5.7070e+00	4.7168e−01	3.6500e−13	3.0439e−01	5.4411e−01



**Table 4** (continued)

		IICS	BCS	CCS	ACS	ICS	NNCS	BHCS	HECS
$f_{13}$	<i>Mean</i>	<b>0</b>	6.1622e−11	3.1990e−24	1.7899e−09	2.7767e−22	2.3176e−24	2.7541e−01	3.4123e+00
	<i>Best</i>	0	1.6431e−11	3.7011e−25	8.2708e−14	3.9638e−23	7.5682e−25	0	2.6992e−12
	<i>Worst</i>	0	1.2503e−10	8.9568e−24	8.0605e−09	1.0681e−21	5.8368e−24	3.0335e+00	3.7010e+01
	<i>SD</i>	0	3.0103e−11	2.0646e−24	2.5233e−09	2.8035e−22	1.1979e−24	6.0988e−01	7.3494e+00
$f_{14}$	<i>Mean</i>	1.7127e−01	3.3363e−01	<b>1.5091e−01</b>	3.1583e−01	2.3970e−01	1.9802e−01	3.2575e−01	4.0319e−01
	<i>Best</i>	1.2699e−01	2.7274e−01	1.2699e−01	2.7275e−01	2.2769e−01	1.7822e−01	2.2769e−01	2.7274e−01
	<i>Worst</i>	2.2769e−01	3.7329e−01	1.7822e−01	3.5186e−01	2.7274e−01	2.2769e−01	3.9610e−01	4.7161e−01
	<i>SD</i>	2.5294e−02	2.2402e−02	2.5551e−02	2.2058e−02	1.9922e−02	2.4228e−02	3.9841e−02	4.7444e−02
$f_{15}$	<i>Mean</i>	1.0319e−03	1.1174e+01	5.8457e+00	1.3973e+01	1.7402e+00	2.7511e+00	<b>2.5535e−16</b>	4.5246e−02
	<i>Best</i>	4.1769e−04	7.7085e+00	3.5273e+00	1.1464e+01	5.1660e−01	1.1364e+00	6.5497e−45	4.6885e−14
	<i>Worst</i>	3.8437e−03	1.3047e+01	7.7856e+00	1.5843e+01	2.9713e+00	5.1818e+00	1.4433e−15	3.4840e−01
	<i>SD</i>	7.8925e−04	1.2460e+00	9.7611e−01	1.1404e+00	5.2155e−01	8.7224e−01	3.3754e−16	7.2348e−02
$f_{16}$	<i>Mean</i>	<b>0</b>	8.0057e−15	1.0583e−27	7.1787e−12	5.1830e−27	1.1272e−26	6.0530e−01	5.2595e−01
	<i>Best</i>	0	2.3315e−15	0	1.6839e−14	0	2.0503e−27	0	2.1266e−16
	<i>Worst</i>	0	2.6559e−14	3.1467e−27	3.9946e−11	1.5264e−26	2.5805e−26	3.2991e+00	2.6656e+00
	<i>SD</i>	0	4.7070e−15	6.7404e−28	9.3992e−12	3.7550e−27	5.5642e−27	8.0578e−01	7.7296e−01
Symbol		+	+	+	+	+	+	+	
Mean Rank	1.56	6.00	3.97	6.75	4.03	3.84	3.91	5.94	

pointed out that when IICS is employed to solve the following model parameter optimization problems, 20 independent calculation runs are conducted to reduce random discrepancy. Correspondingly, the mean predicted values are utilized for the following hidden neuron number selection, as well as the model prediction performance exhibition and comparison.

For each of the above two empirical models, the activation functions for the hidden layer and the output layer are the sigmoid function and the linear function, respectively. The optimal numbers of hidden neurons for  $\hat{f}_{arc}$  and  $\hat{f}_{sur}$  are 3 and 5, respectively, determined by trial and error; that is to say, the selected topologies of these two empirical models are 2–3–1 and 3–5–1, respectively. After the topologies of  $\hat{f}_{arc}$  and  $\hat{f}_{sur}$  are selected, all the 437 heats of modeling data are utilized for determining the model parameters with the methodology depicted in Sect. 4, to obtain the overall hybrid temperature prediction model of molten steel.

Then, the 100 heats of testing data are utilized to evaluate the performance of the proposed hybrid prediction model. Figure 7 shows the final molten steel temperature predicted by the developed hybrid model. Figure 7 shows that this model can predict the temperature with high accuracy. Out of these prediction results, the absolute error in 91% of the cases is lower than 5 °C (desirable value),

and in 95% of the cases, it is lower than 7 °C (tolerable value), and only in 2% of the cases is absolute error higher than 10 °C. This demonstrates the effectiveness of the hybrid prediction model, with the proposed indirect training method for its empirical part.

To demonstrate the excellent prediction ability of the proposed hybrid model, this article also develops an empirical prediction model based on the above selected 437 heats of production data. To be fair to the comparison, this empirical model is also established utilizing a SLFN and its parameters (namely the network's weights and thresholds) are determined by IICS. The input layer of this SLFN-based empirical prediction model has eight neurons. They are the initial molten steel temperature, total power consumption, ladle state, heat effect of additions, total argon consumption, weight of molten steel, refining time, and energy change of cooling water in the water-cooled cover. The hidden layer has 13 neurons (determined by trial and error), and the output layer has one neuron (the final molten steel temperature). Figure 8 shows the results predicted by the empirical model. For ease of comparison, the prediction errors (*PE*) of the proposed hybrid model and empirical model, as well as the differences between the prediction errors (*D<sub>PE</sub>*) of these models are presented in Fig. 9a, b respectively. Herein, the differences in more details are the results obtained by subtracting the absolute

Table 5 Test results on CEC 2017 benchmark functions

	IICS	BCS	CCS	ACS	ICS	NNCS	BHCS	HECS	HFPSO	ELSHADE-SPACMA	HS-ES	ISCA
$f_1$	Mean	1.65e+02	1.59e+04	1.67e+03	8.37e+03	3.58e+03	3.02e+03	2.83e+03	5.35e+03	3.35e+07	0.00e+00	8.46e+07
	Best	9.38e-01	2.77e+03	8.07e-01	4.68e+02	1.93e+00	2.31e+00	2.83e+02	9.99e+02	8.56e+00	0.00e+00	4.82e+06
	Worst	5.04e+02	4.01e+04	8.54e+03	5.32e+04	1.59e+04	1.01e+04	2.05e+04	3.14e+04	8.55e+08	3.05e-08	6.13e+08
	SD	1.60e+02	8.91e+03	1.91e+03	9.62e+03	3.99e+03	2.85e+03	4.12e+03	6.39e+03	1.66e+08	0.00e+00	1.04e+08
$f_3$	Mean	6.09e+04	1.52e+05	8.92e+04	1.34e+05	1.41e+05	1.04e+05	5.03e-05	2.63e+00	4.18e+02	0.00e+00	1.74e+04
	Best	4.49e+04	1.23e+05	5.53e+04	9.31e+04	1.03e+05	6.24e+04	2.39e-08	5.70e-01	4.35e-06	0.00e+00	8.93e+03
	Worst	6.86e+04	1.86e+05	1.07e+05	1.65e+05	1.65e+05	1.27e+05	6.21e-04	2.10e+01	8.08e+03	1.89e-08	2.80e+04
	SD	6.34e+03	1.49e+04	1.01e+04	1.61e+04	1.44e+04	1.50e+04	1.29e-04	3.73e+00	1.44e+03	0.00e+00	4.33e+03
$f_4$	Mean	4.87e+01	5.00e+01	9.00e+01	9.11e+01	8.95e+01	6.16e+01	1.08e+02	1.91e+02	1.04e+02	3.19e+01	1.65e+02
	Best	2.85e+01	2.98e+01	3.02e+01	4.30e+01	3.28e+01	2.86e+01	7.15e+01	1.49e+02	3.92e+00	0.00e+00	7.76e+01
	Worst	7.32e+01	1.17e+02	1.58e+02	1.57e+02	1.21e+02	1.23e+02	1.43e+02	2.29e+02	2.21e+02	1.14e+02	2.45e+02
	SD	1.89e+01	1.93e+01	2.91e+01	2.72e+01	2.35e+01	2.85e+01	2.71e+01	2.73e+01	5.14e+01	4.29e+01	3.75e+01
$f_5$	Mean	1.20e+02	4.40e+02	2.58e+02	2.82e+02	1.70e+02	2.19e+02	2.26e+02	1.45e+02	2.69e+02	1.35e+00	1.53e+02
	Best	9.88e+01	3.57e+02	1.94e+02	2.18e+02	1.31e+02	1.45e+02	1.96e+02	1.25e+02	1.89e+02	0.00e+00	1.01e+02
	Worst	1.27e+02	5.09e+02	3.11e+02	3.39e+02	2.03e+02	2.70e+02	3.20e+02	1.92e+02	3.73e+02	3.98e+00	2.03e+02
	SD	7.31e+00	3.88e+01	2.09e+01	2.54e+01	1.45e+01	2.89e+01	2.97e+01	1.71e+01	4.88e+01	1.07e+00	2.29e+01
$f_6$	Mean	0.00e+00	4.24e+01	3.75e+00	8.28e+00	7.24e-01	1.59e-01	1.53e+01	8.59e+00	3.39e+01	4.50e-07	3.82e-01
	Best	0.00e+00	2.61e+01	2.26e+00	2.68e+00	3.84e-01	5.26e-02	1.17e+01	6.68e+00	1.45e+01	0.00e+00	1.53e-01
	Worst	0.00e+00	5.19e+01	7.10e+00	2.60e+01	1.08e+00	3.73e-01	2.57e+01	1.38e+01	5.72e+01	2.06e-06	9.62e-01
	SD	0.00e+00	5.63e+00	1.10e+00	3.89e+00	1.52e-01	7.09e-02	3.51e+00	1.73e+00	1.07e+01	4.70e-07	1.69e-01
$f_7$	Mean	1.64e+02	4.83e+02	3.62e+02	3.60e+02	2.28e+02	3.24e+02	3.92e+02	2.47e+02	3.27e+02	5.51e+01	2.42e+02
	Best	1.39e+02	4.11e+02	2.89e+02	3.05e+02	1.91e+02	2.40e+02	3.26e+02	2.20e+02	2.15e+02	5.38e+01	1.91e+02
	Worst	1.74e+02	5.81e+02	4.07e+02	4.16e+02	2.63e+02	3.77e+02	5.26e+02	3.05e+02	4.48e+02	5.63e+01	2.95e+02
	SD	8.14e+00	3.87e+01	2.38e+01	2.68e+01	1.54e+01	2.88e+01	4.74e+01	2.05e+01	5.98e+01	6.08e-01	2.50e+01
$f_8$	Mean	1.18e+02	4.31e+02	2.47e+02	2.78e+02	1.68e+02	2.22e+02	2.08e+02	1.60e+02	2.70e+02	1.60e+00	1.54e+02
	Best	9.69e+01	3.15e+02	1.84e+02	1.79e+02	1.26e+02	1.34e+02	1.88e+02	1.32e+02	1.87e+02	0.00e+00	8.14e+01
	Worst	1.31e+02	5.21e+02	2.98e+02	3.27e+02	2.06e+02	2.84e+02	2.57e+02	2.12e+02	3.58e+02	4.97e+00	2.05e+02
	SD	9.00e+00	3.99e+01	2.46e+01	3.07e+01	1.80e+01	2.64e+01	1.86e+01	1.91e+01	3.80e+01	1.26e+00	2.72e+01
$f_9$	Mean	2.20e+03	2.98e+04	6.35e+03	9.02e+03	1.97e+03	1.03e+04	3.17e+03	1.45e+03	8.41e+03	0.00e+00	1.56e+02
	Best	1.42e+03	1.06e+04	1.84e+03	4.09e+03	7.17e+02	3.39e+03	2.13e+03	1.05e+03	1.74e+03	0.00e+00	8.96e+00
	Worst	2.91e+03	3.85e+04	1.04e+04	1.97e+04	3.48e+03	1.73e+04	6.21e+03	2.24e+03	1.88e+04	0.00e+00	5.89e+02
	SD	5.08e+02	4.92e+03	1.73e+03	3.65e+03	6.04e+02	2.43e+03	9.87e+02	3.19e+02	2.81e+03	0.00e+00	1.21e+02
$f_{10}$	Mean	5.59e+03	6.59e+03	7.78e+03	8.09e+03	6.58e+03	6.46e+03	1.16e+04	5.75e+03	6.43e+03	4.53e+02	6.40e+03
	Best	4.81e+03	6.03e+03	6.82e+03	6.81e+03	5.89e+03	5.55e+03	1.03e+04	5.13e+03	4.67e+03	1.25e+02	4.79e+03
	Worst	5.93e+03	7.27e+03	9.00e+03	8.77e+03	7.19e+03	7.15e+03	1.33e+04	7.45e+03	8.17e+03	1.59e+03	7.71e+03
	SD	2.91e+02	3.10e+02	4.77e+02	3.81e+02	2.96e+02	3.84e+02	8.77e+02	4.75e+02	8.50e+02	4.09e+02	7.85e+02

Table 5 (continued)

	IICS	BCS	CCS	ACS	ICS	NNCS	BHCS	HECS	HFPSSO	ELSHADE-SPACMA	HS-ES	ISCA
$f_{11}$	Mean	1.12e+02	2.42e+02	2.09e+02	2.19e+02	1.01e+02	1.43e+02	2.26e+02	2.23e+02	2.62e+01	2.30e+01	2.07e+02
	Best	9.25e+01	1.88e+02	1.68e+02	1.76e+02	7.92e+01	9.23e+01	1.82e+02	8.79e+01	2.03e+01	1.14e+08	1.15e+02
	Worst	1.24e+02	2.84e+02	2.48e+02	2.65e+02	1.52e+02	1.85e+02	2.81e+02	3.50e+02	3.31e+01	2.62e+01	3.76e+02
	SD	8.10e+00	2.17e+01	1.80e+01	2.40e+01	1.48e+01	1.96e+01	3.17e+01	5.36e+01	3.76e+00	3.98e+00	5.18e+01
$f_{12}$	Mean	7.53e+05	3.16e+06	6.91e+06	9.22e+06	2.75e+06	3.14e+06	7.86e+04	5.05e+05	1.36e+03	1.25e+02	1.53e+07
	Best	2.87e+05	1.32e+06	3.51e+06	5.15e+06	4.61e+05	8.03e+05	3.72e+04	2.06e+05	4.03e+02	2.44e+00	4.07e+06
	Worst	9.53e+05	5.10e+06	9.87e+06	1.52e+07	5.31e+06	5.72e+06	1.97e+05	1.38e+06	2.28e+03	5.19e+02	4.87e+07
	SD	1.86e+05	7.99e+05	1.45e+06	2.35e+06	1.04e+06	1.21e+06	4.31e+04	3.25e+05	3.42e+02	1.30e+02	8.90e+06
$f_{13}$	Mean	8.28e+02	9.49e+03	2.50e+03	7.83e+03	8.59e+03	2.15e+03	7.80e+03	7.35e+03	3.68e+01	4.78e+01	6.67e+04
	Best	5.72e+02	1.59e+03	3.72e+02	4.82e+02	3.46e+02	4.35e+02	3.11e+03	2.82e+03	1.07e+01	2.92e+01	2.04e+04
	Worst	1.04e+03	2.87e+04	1.17e+04	3.02e+04	3.71e+04	5.26e+03	1.93e+04	4.28e+04	6.71e+01	8.62e+01	2.46e+05
	SD	1.31e+02	6.63e+03	2.33e+03	7.63e+03	9.58e+03	1.26e+03	3.81e+03	8.41e+03	1.72e+01	2.42e+01	4.29e+04
$f_{14}$	Mean	7.73e+04	1.06e+05	1.27e+05	1.62e+05	8.72e+04	1.72e+05	5.46e+02	4.41e+02	3.07e+01	1.53e+01	1.32e+05
	Best	9.46e+03	1.18e+04	2.65e+04	5.72e+04	2.79e+04	4.10e+04	4.00e+02	3.67e+02	2.52e+01	3.79e+04	2.08e+04
	Worst	1.01e+05	1.95e+05	2.97e+05	4.89e+05	2.01e+05	3.42e+05	9.26e+02	5.60e+02	4.25e+01	2.24e+01	2.81e+05
	SD	2.19e+04	3.76e+04	6.05e+04	8.48e+04	3.51e+04	7.55e+04	1.37e+02	1.38e+05	3.95e+00	8.64e+00	6.77e+04
$f_{15}$	Mean	3.37e+02	8.12e+02	8.76e+02	1.38e+03	7.92e+02	4.02e+02	8.99e+03	4.49e+03	2.28e+01	1.67e+01	1.64e+04
	Best	1.30e+02	4.01e+02	1.53e+02	2.51e+02	9.93e+01	2.12e+02	4.00e+03	3.51e+02	1.94e+01	1.34e+00	2.79e+03
	Worst	4.85e+02	1.54e+03	1.84e+03	1.20e+04	2.43e+03	9.47e+02	1.82e+04	1.86e+04	2.91e+01	1.94e+01	2.37e+04
	SD	8.42e+01	2.19e+02	4.11e+02	2.10e+03	5.45e+02	1.12e+02	4.41e+03	6.82e+03	2.20e+00	3.74e+00	5.52e+03
$f_{16}$	Mean	8.83e+02	1.51e+03	1.30e+03	1.36e+03	1.09e+03	1.15e+03	1.72e+03	1.55e+03	4.15e+02	5.31e+02	1.12e+03
	Best	5.36e+02	1.08e+03	9.71e+02	7.60e+02	6.69e+02	5.42e+02	1.42e+03	8.23e+02	1.29e+02	1.29e+02	6.36e+02
	Worst	1.07e+03	1.94e+03	1.60e+03	1.77e+03	1.48e+03	1.58e+03	2.18e+03	2.80e+03	8.66e+02	9.35e+02	1.48e+03
	SD	1.37e+02	1.89e+02	1.46e+02	2.13e+02	1.63e+02	1.96e+02	2.32e+02	4.11e+02	1.77e+02	2.04e+02	1.84e+02
$f_{17}$	Mean	6.03e+02	1.02e+03	7.83e+02	9.05e+02	7.38e+02	7.82e+02	1.20e+03	1.22e+03	2.30e+02	2.78e+02	7.21e+02
	Best	4.07e+02	7.05e+02	4.58e+02	4.71e+02	3.94e+02	3.32e+02	9.75e+02	5.19e+02	3.48e+01	1.50e+02	4.45e+02
	Worst	6.83e+02	1.37e+03	1.13e+03	1.13e+03	1.02e+03	1.11e+03	1.90e+03	1.75e+03	5.09e+02	6.78e+02	1.05e+03
	SD	5.95e+01	1.61e+02	1.29e+02	1.53e+02	1.39e+02	1.70e+02	2.14e+02	3.07e+02	9.68e+01	1.44e+02	1.36e+02
$f_{18}$	Mean	3.03e+05	6.66e+05	7.35e+05	9.33e+05	6.34e+05	5.88e+05	9.76e+03	2.85e+04	2.51e+01	2.06e+01	8.78e+05
	Best	1.63e+05	1.95e+05	1.75e+05	3.04e+05	2.85e+05	1.68e+05	4.12e+03	1.65e+04	2.16e+01	1.54e+00	7.40e+04
	Worst	4.14e+05	1.11e+06	1.46e+06	2.26e+06	1.37e+06	1.42e+06	2.62e+04	7.67e+05	3.43e+01	2.20e+01	3.01e+06
	SD	7.18e+04	2.26e+05	2.99e+05	3.88e+05	2.43e+05	2.56e+05	5.01e+03	1.24e+05	2.56e+00	2.72e+00	5.93e+05
$f_{19}$	Mean	4.31e+02	3.77e+02	1.08e+03	1.47e+03	6.39e+02	4.04e+02	1.20e+03	5.70e+03	1.44e+01	7.58e+00	2.33e+04
	Best	1.23e+02	1.69e+02	8.62e+01	1.08e+02	1.14e+02	7.36e+01	2.74e+02	1.83e+02	8.46e+00	3.68e+00	5.04e+02
	Worst	7.02e+02	1.49e+03	5.06e+03	8.95e+03	2.53e+03	4.32e+03	1.10e+04	7.05e+04	1.92e+01	2.20e+01	4.42e+04
	SD	1.56e+02	2.11e+02	1.04e+03	1.85e+03	5.22e+02	6.50e+02	2.48e+03	1.20e+04	2.31e+00	4.28e+00	1.32e+04

Table 5 (continued)

		IICS	BCS	CCS	ACS	ICS	NNCS	BHCS	HECS	HFPSO	ELSHADE-SPACMA	HS-ES	ISCA
$f_{20}$	Mean	4.52e+02	9.30e+02	5.78e+02	7.60e+02	5.59e+02	6.47e+02	8.21e+02	8.25e+02	7.87e+02	1.08e+02	4.45e+01	5.61e+02
	Best	2.34e+02	5.59e+02	2.42e+02	4.61e+02	1.84e+02	3.55e+02	5.43e+02	6.95e+02	4.36e+02	3.66e+01	3.73e+00	2.13e+02
	Worst	5.56e+02	1.22e+03	8.47e+02	1.09e+03	8.38e+02	9.61e+02	1.26e+03	1.21e+03	1.48e+03	3.01e+02	2.45e+02	8.62e+02
	SD	8.12e+01	1.62e+02	1.32e+02	1.31e+02	1.40e+02	1.47e+02	2.19e+02	1.21e+02	2.33e+02	7.31e+01	5.33e+01	1.39e+02
$f_{21}$	Mean	3.14e+02	6.24e+02	4.51e+02	4.67e+02	3.64e+02	4.18e+02	3.63e+02	3.41e+02	4.65e+02	2.42e+02	2.05e+02	3.48e+02
	Best	2.87e+02	5.44e+02	3.87e+02	4.18e+02	3.23e+02	3.54e+02	3.40e+02	3.19e+02	3.73e+02	2.23e+02	2.00e+02	3.01e+02
	Worst	3.25e+02	6.97e+02	5.02e+02	5.05e+02	4.05e+02	4.71e+02	4.17e+02	3.86e+02	6.06e+02	2.62e+02	2.10e+02	4.06e+02
	SD	9.75e+00	3.14e+01	2.30e+01	2.34e+01	1.71e+01	2.12e+01	2.11e+01	1.97e+01	5.11e+01	9.52e+00	1.63e+00	2.42e+01
$f_{22}$	Mean	3.68e+03	7.24e+03	6.17e+03	8.61e+03	7.03e+03	6.85e+03	1.21e+04	6.34e+03	6.82e+03	7.16e+02	1.00e+02	6.26e+03
	Best	1.61e+02	4.90e+02	1.00e+02	7.24e+03	6.23e+03	2.22e+02	1.02e+04	5.66e+03	1.00e+02	1.00e+02	1.00e+02	1.90e+02
	Worst	6.44e+03	8.16e+03	9.02e+03	9.48e+03	7.77e+03	7.82e+03	1.33e+04	7.76e+03	9.03e+03	4.85e+03	1.00e+02	8.26e+03
	SD	2.89e+03	1.02e+03	3.62e+03	3.99e+02	3.37e+02	1.04e+03	8.43e+02	5.10e+02	2.08e+03	1.44e+03	0.00e+00	1.94e+03
$f_{23}$	Mean	5.52e+02	8.56e+02	6.88e+02	6.84e+02	5.97e+02	6.50e+02	6.84e+02	6.11e+02	9.25e+02	4.62e+02	4.24e+02	5.97e+02
	Best	5.19e+02	7.75e+02	5.54e+02	6.24e+02	5.62e+02	5.93e+02	6.33e+02	5.87e+02	6.76e+02	4.35e+02	4.05e+02	5.24e+02
	Worst	5.64e+02	9.52e+02	7.56e+02	7.34e+02	6.29e+02	6.97e+02	8.36e+02	6.92e+02	1.34e+03	4.89e+02	4.47e+02	6.57e+02
	SD	1.00e+01	4.31e+01	3.40e+01	2.34e+01	1.28e+01	2.58e+01	4.68e+01	2.12e+01	1.48e+02	1.39e+01	8.29e+00	2.91e+01
$f_{24}$	Mean	6.91e+02	9.54e+02	7.59e+02	7.20e+02	6.54e+02	7.53e+02	7.62e+02	7.00e+02	9.93e+02	5.34e+02	4.89e+02	6.72e+02
	Best	6.49e+02	8.54e+02	6.72e+02	6.74e+02	6.10e+02	6.60e+02	6.95e+02	6.66e+02	8.34e+02	5.14e+02	4.78e+02	6.07e+02
	Worst	7.06e+02	1.06e+03	8.27e+02	7.66e+02	6.92e+02	8.37e+02	8.36e+02	7.65e+02	1.34e+03	5.62e+02	4.95e+02	7.48e+02
	SD	1.48e+01	4.56e+01	2.93e+01	2.27e+01	1.72e+01	3.36e+01	4.62e+01	2.77e+01	1.14e+02	9.14e+00	2.73e+00	3.18e+01
$f_{25}$	Mean	5.26e+02	5.08e+02	5.18e+02	5.10e+02	4.96e+02	5.08e+02	5.90e+02	6.01e+02	5.55e+02	4.81e+02	5.49e+02	5.52e+02
	Best	4.61e+02	4.69e+02	4.72e+02	4.64e+02	4.72e+02	4.68e+02	5.64e+02	5.85e+02	4.61e+02	4.77e+02	4.71e+02	4.99e+02
	Worst	5.49e+02	5.34e+02	5.43e+02	5.37e+02	5.26e+02	5.42e+02	6.20e+02	6.28e+02	6.14e+02	4.92e+02	5.65e+02	6.24e+02
	SD	1.95e+01	1.64e+01	1.45e+01	1.85e+01	1.63e+01	1.75e+01	1.66e+01	1.22e+01	3.95e+01	2.80e+00	1.89e+01	2.29e+01
$f_{26}$	Mean	2.13e+03	2.64e+03	2.91e+03	3.60e+03	2.83e+03	1.59e+03	4.98e+03	3.50e+03	3.10e+03	1.34e+03	6.24e+02	2.84e+03
	Best	6.34e+02	7.26e+02	3.00e+02	1.57e+03	1.10e+03	3.00e+02	4.26e+03	3.13e+03	3.00e+02	1.08e+03	4.00e+02	2.37e+03
	Worst	2.50e+03	6.15e+03	4.43e+03	4.35e+03	3.22e+03	3.78e+03	6.56e+03	4.01e+03	7.80e+03	1.64e+03	9.41e+02	3.48e+03
	SD	5.16e+02	1.88e+03	1.33e+03	5.06e+02	2.96e+02	1.28e+03	5.78e+02	2.59e+02	2.90e+03	1.38e+02	1.58e+02	2.66e+02
$f_{27}$	Mean	5.74e+02	7.46e+02	7.52e+02	6.21e+02	5.73e+02	6.52e+02	1.04e+03	7.75e+02	8.21e+02	5.10e+02	5.87e+02	5.70e+02
	Best	5.27e+02	6.71e+02	6.34e+02	5.42e+02	5.44e+02	5.82e+02	9.33e+02	7.09e+02	6.10e+02	4.80e+02	5.48e+02	5.34e+02
	Worst	5.88e+02	8.48e+02	8.55e+02	7.00e+02	6.09e+02	7.24e+02	1.22e+03	9.08e+02	1.21e+03	5.28e+02	6.12e+02	6.42e+02
	SD	1.20e+01	4.03e+01	4.01e+01	2.95e+01	1.85e+01	3.05e+01	8.70e+01	5.80e+01	1.42e+02	9.52e+00	1.54e+01	2.52e+01
$f_{28}$	Mean	5.00e+02	4.63e+02	4.61e+02	4.62e+02	4.59e+02	4.61e+02	5.12e+02	5.66e+02	4.98e+02	4.60e+02	5.02e+02	5.25e+02
	Best	4.78e+02	4.60e+02	4.59e+02	4.59e+02	4.59e+02	4.59e+02	4.97e+02	5.43e+02	4.59e+02	4.59e+02	4.59e+02	4.76e+02
	Worst	5.08e+02	4.82e+02	5.00e+02	4.89e+02	4.60e+02	4.76e+02	5.99e+02	6.06e+02	5.56e+02	5.08e+02	5.08e+02	5.74e+02
	SD	9.97e+00	3.63e+00	5.90e+00	4.49e+00	2.42e-01	3.57e+00	2.39e+01	1.79e+01	2.29e+01	6.84e+00	1.23e+01	2.33e+01

Table 5 (continued)

	IICS	BCS	CCS	ACS	ICS	NNCS	BHCS	HECS	HFPSSO	ELSHADE-SPACMA	HS-ES	ISCA
$f_{29}$												
Mean	6.10e+02	1.36e+03	1.24e+03	1.28e+03	8.37e+02	9.16e+02	1.57e+03	1.07e+03	1.49e+03	3.58e+02	4.77e+02	8.32e+02
Best	4.42e+02	1.09e+03	8.63e+02	8.63e+02	5.64e+02	5.95e+02	1.39e+03	8.76e+02	7.78e+02	3.27e+02	3.05e+02	5.92e+02
Worst	6.81e+02	1.76e+03	1.59e+03	1.65e+03	1.03e+03	1.27e+03	1.95e+03	1.43e+03	2.14e+03	4.10e+02	8.43e+02	1.13e+03
SD	6.51e+01	1.38e+02	1.41e+02	1.73e+02	1.13e+02	1.40e+02	1.57e+02	1.39e+02	3.63e+02	1.78e+01	1.61e+02	1.33e+02
$f_{30}$												
Mean	6.50e+05	1.29e+06	3.49e+06	1.34e+06	8.33e+05	8.61e+05	6.51e+05	7.51e+05	1.11e+06	5.97e+05	6.03e+05	3.76e+06
Best	6.18e+05	7.67e+05	9.51e+05	6.82e+05	6.72e+05	7.02e+05	5.99e+05	6.49e+05	6.31e+05	5.79e+05	5.81e+05	2.12e+06
Worst	6.73e+05	1.72e+06	6.04e+06	2.59e+06	1.07e+06	1.08e+06	8.88e+05	9.33e+05	5.07e+06	6.63e+05	6.95e+05	1.00e+07
SD	1.56e+04	2.20e+05	1.10e+06	4.59e+05	9.59e+04	9.19e+04	6.40e+04	9.51e+04	6.15e+05	2.38e+04	2.46e+04	1.33e+06
Symbol	+	+	+	+	+	+	+	+	-	-	+	+
Mean Rank	4.02	7.74	9.05	5.64	6.45	8.67	7.17	9.21	1.69	1.95	7.19	

prediction errors of the empirical model from that of the hybrid model. Thus, obviously the  $D\_PE$  value can indicate whether the prediction performance of the hybrid model is better than (i.e.,  $D\_PE < 0$ ), ties (i.e.,  $D\_PE = 0$ ), or worse than (i.e.,  $D\_PE > 0$ ) that of the empirical model on the corresponding heat. In addition, four performance evaluation indices are used for quantitative comparison. They are the root mean square error (RMSE), mean absolute error (MAE), mean relative error (MRE), and accuracy rate (AR) which is defined as

$$AR = \frac{N_a}{N_t} \cdot 100\% \quad (35)$$

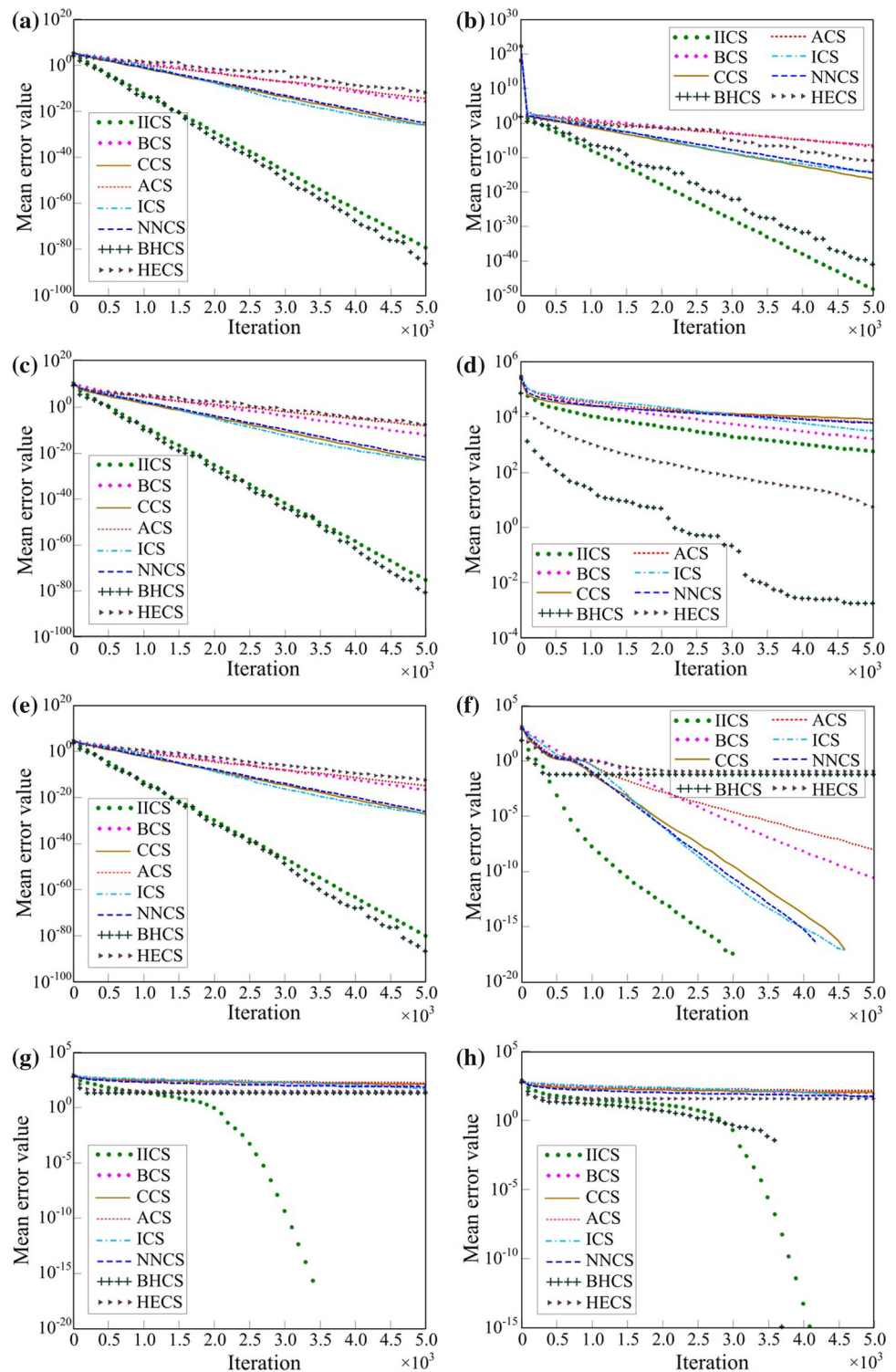
where  $N_a$  is the number of heats with absolute prediction errors not higher than 5 °C, and  $N_t$  is the number of total testing heats. The calculation results with respect to these four indices for the above two models are listed in Table 6.

From Figs. 7–9, it can be observed that both the hybrid model and the empirical one could predict the molten steel temperature with certain accuracy, while the prediction values given by the former are much closer to the measured values than those given by the latter. Furthermore, as can be observed from the data in Table 6, the proposed hybrid model predicts significantly better than the empirical one. Compared with the empirical model, the RMSE, MAE, and MRE of the proposed hybrid model are respectively lower by 33.11%, 30.81%, and 30.77%; while the AR of the proposed hybrid model exceeds 90%, an 18.18% improvement over the empirical one. These demonstrate the excellent prediction performance of the proposed hybrid model in a practical application. From these observations and comparisons above, it can be concluded that the proposed hybrid model is a promising predictor for the molten steel temperature.

Moreover, to confront the search capability of IICS with some other widely used intelligent algorithms in model parameter optimization, GA [12], PSO [13], DE [14], ACO [15], and BCS are employed to solve the same PO problem with the topologies of  $\hat{f}_{arc}$  and  $\hat{f}_{sur}$  being 2–3–1 and 3–5–1 respectively, and the 437 heats of modeling data. In addition, it is interesting to investigate the performance of the two winners of the CEC 2018 competition, i.e., HSES and ELSHADE-SPACMA, on this PO problem. The parameters of these four new selected algorithms are set according to the respective studies. Specifically, in GA the BLX- $\alpha$  Crossover is used, and the crossover probability  $p_c$ , mutation probability  $p_m$ , and parameter  $\alpha$  for BLX are set to 0.85, 0.02, and 2.0 respectively. In PSO the inertia weight  $\omega$ , cognitive acceleration coefficient  $c_1$ , and social acceleration coefficient  $c_2$  are taken as 0.5, 2.0, and 1.0, respectively. In DE the scale factor  $F$  and crossover probability  $P_{xover}$  are both set to 0.5, and the mutation operator is DE/best/1. In ACO the selection parameter  $q_0$ ,



**Fig. 6** Mean convergence characteristics of eight CS-based algorithms on 16 classical benchmark functions. **a**  $f_1$ , **b**  $f_2$ , **c**  $f_3$ , **d**  $f_4$ , **e**  $f_5$ , **f**  $f_6$ , **g**  $f_7$ , **h**  $f_8$ , **i**  $f_9$ , **j**  $f_{10}$ , **k**  $f_{11}$ , **l**  $f_{12}$ , **m**  $f_{13}$ , **n**  $f_{14}$ , **o**  $f_{15}$ , and **p**  $f_{16}$

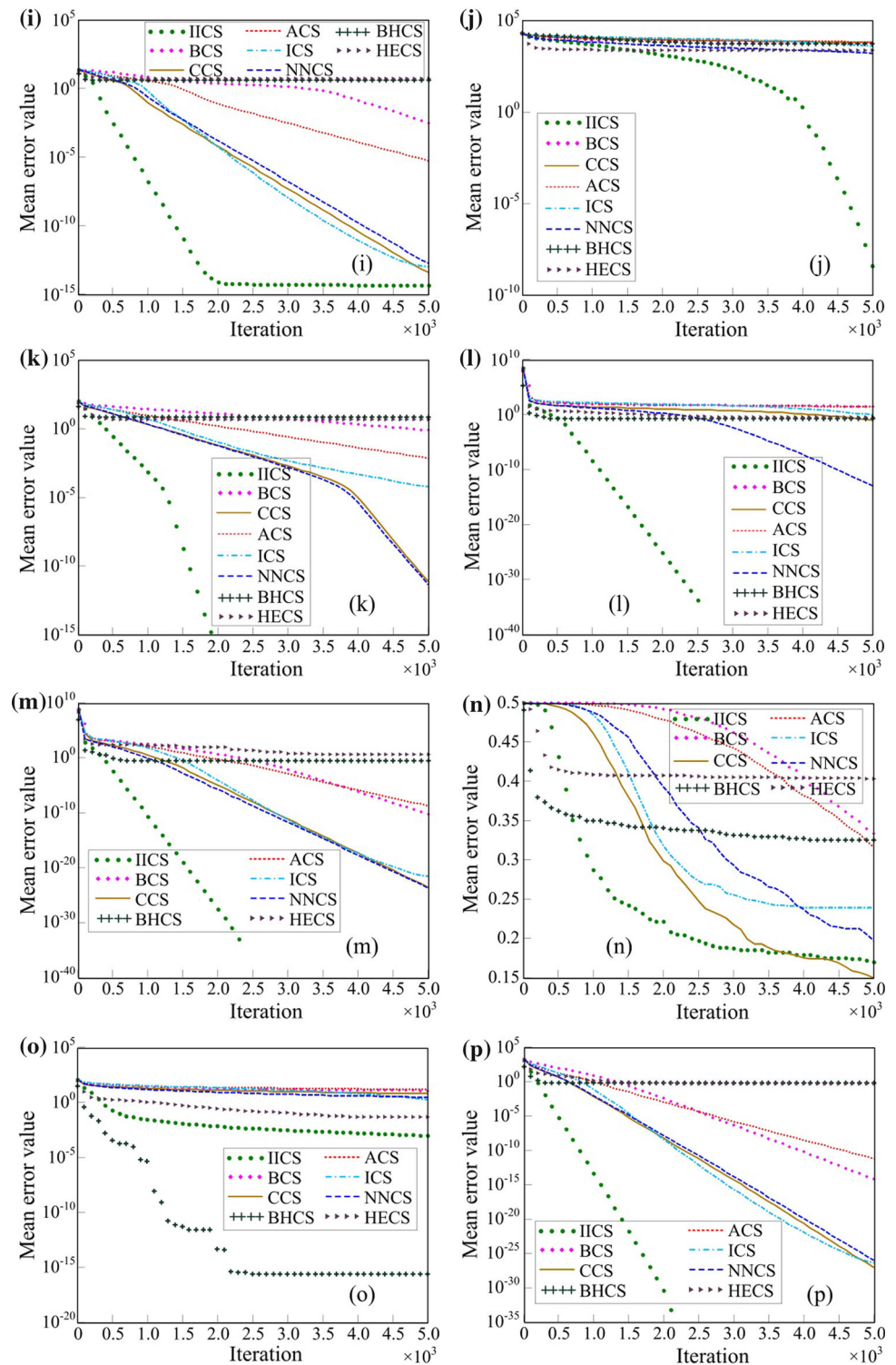


two weighting parameters  $\alpha$  and  $\beta$ , and two updating parameters  $Q$  and  $\gamma$  are respectively set to 0.85, 1, 4, 0.1, and 0.7. The parameters of IICS, BCS, ELSHADE-SPACMA, and HSES are in common with those used in Sect. 5.1. The size of population  $N_p$  is set to be the number of model parameters to be estimated (i.e.,  $D = 39$ ) for IICS,

GA, PSO, DE, ACO, and BCS, while the settings of  $N_p$  for ELSHADE-SPACMA and HSES are consistent with the original studies. To be fair, the maximum number of fitness function evaluations in all 20 runs is equal to  $10,000 \times D$ .

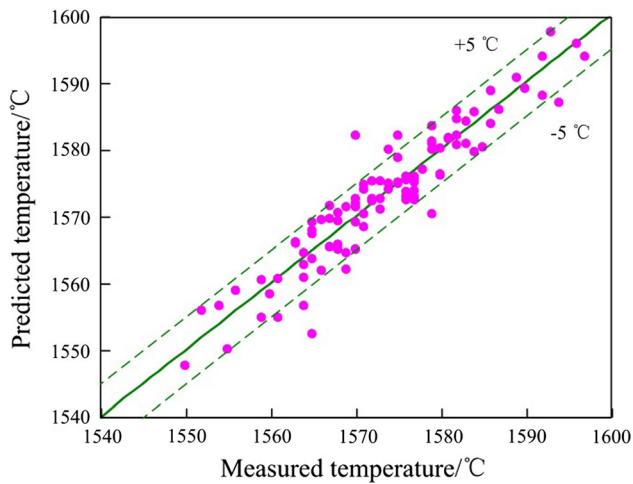
Table 7 gives the results of the eight algorithms on the PO problem in 20 independent runs, and the best results are

Fig. 6 continued

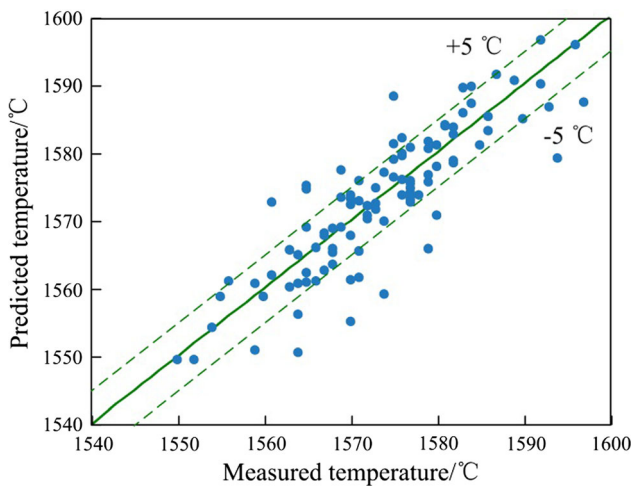


shown in boldface for clarity. The columns headed ‘Mean’, ‘Min’, ‘Max’, and ‘SD’ show the mean, minimum, maximum, and standard deviation values of the fitness function defined by Eq. (20) for each algorithm. As can be seen from Table 7, in terms of the mean fitness function value both BCS and IICS give better results than the other four

widely-used model parameter optimization algorithms, revealing that CS is relatively more suitable for solving the PO problem involved in this study. It can also be seen that IICS performs better than ELSHADE-SPACMA and HSES, which indicates that IICS is an efficient algorithm for solving the PO problem. Furthermore, it can be

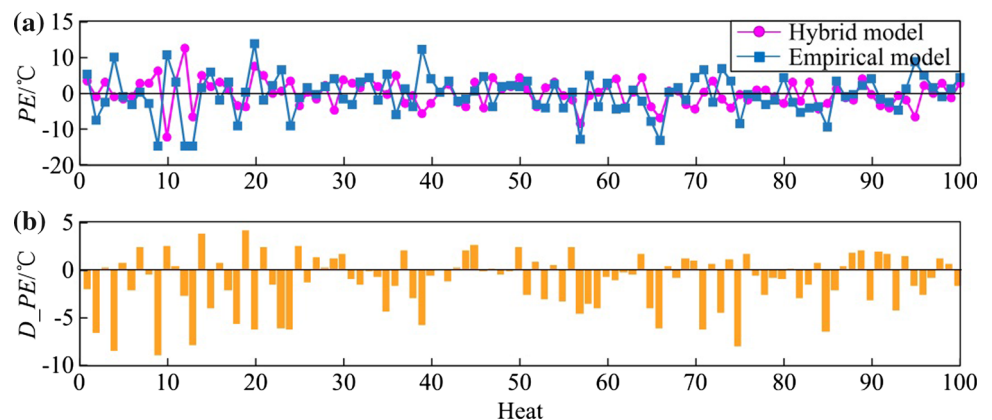


**Fig. 7** Predicted results of the molten steel temperature by the proposed hybrid model



**Fig. 8** Prediction results of the molten steel temperature by the empirical model

**Fig. 9** Comparison of prediction errors between the proposed hybrid model and empirical model



observed that IICS produces much better optimization results than BCS. Figure 10 illustrates the convergence progress of IICS and BCS on the PO problem. It can be found from Fig. 10 that IICS has a better search accuracy and a higher convergence speed, which demonstrates again that the proposed IICS algorithm has greatly enhanced the performance of BCS.

## 6 Conclusions

A hybrid model for the prediction of molten steel temperature in LF is proposed. In the proposed hybrid prediction model, two SLFN-based empirical models are incorporated within the structure of a mechanistic thermal model, to represent the unknown functions in the mechanistic thermal model. The primary difference between the proposed hybrid prediction model and existing ones is that its empirical part is not trained in the traditional direct way since the target outputs of the two empirical models are unavailable in advance. In the proposed approach, the empirical part is trained indirectly with the readily available temperature measurements of molten steel but not the barely accessible target outputs of this part, which means the hybrid prediction model with its empirical part trained by the proposed indirect method has more extensive application range when compared to existing ones. Application results on the production data from a 300 t LF at Baoshan Iron & Steel Co., Ltd, show the effectiveness and superiority of the proposed hybrid prediction model.

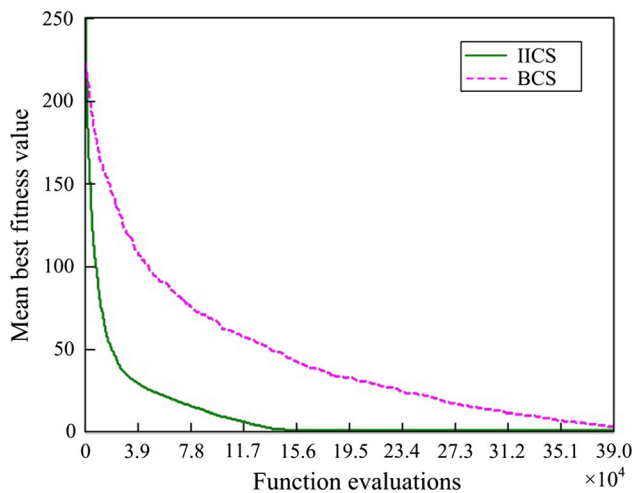
Another main innovation of this article is the development of the information interaction-enhanced CS (IICS), which is used to optimize the parameters in the empirical part so as to complete the development of the proposed hybrid prediction model. One of the problems with BCS and many of its variants is that the information interaction among cuckoos is lacking in the search process, which

**Table 6** Comparison of test performance between the proposed hybrid model and empirical model

Model	RMSE (°C)	MAE (°C)	MRE (%)	AR (%)
Hybrid model	3.6205	2.8348	0.1802	91
Empirical model	5.4123	4.0970	0.2603	77

**Table 7** Results of IICS, GA, PSO, DE, ACO, BCS, ELSHADE-SPACMA, and HSES on the PO problem

Algorithm	Fitness value			
	Mean	Min	Max	SD
IICS	<b>2.0114</b>	<b>1.7906</b>	<b>2.9127</b>	<b>0.2713</b>
GA	5.2436	3.5509	10.2785	1.5469
PSO	5.1957	3.4996	10.5761	1.7069
DE	4.7259	3.2485	8.8003	1.1419
ACO	5.8124	4.1577	8.9595	0.6557
BCS	4.1357	3.0184	5.4039	0.4922
ELSHADE-SPACMA	3.9632	2.8097	6.1785	0.5469
HSES	3.6147	3.0589	5.3270	0.4734

**Fig. 10** Convergence curves of IICS and BCS on the PO problem

would decrease their search performance considerably. In order to overcome this problem, an information interaction-enhanced mechanism is proposed and employed in IICS. The optimization results on the model parameter estimation problem and two benchmark sets (16 classical benchmark functions and 29 CEC 2017 benchmark functions) indicate that IICS has distinct advantages over its competitive algorithms (except ELSHADE-SPACMA and HSES) on these optimization problems. When compared with the winners of the CEC 2018 competition, i.e., ELSHADE-

SPACMA and HSES, the performance of IICS is found to be inferior to the two top algorithms on the CEC 2017 benchmark set, but it produces better results on the parameter optimization problem involved in this study. Despite its promising performance, the proposed IICS still has limitations. First of all, compared with BCS, one more parameter (i.e.,  $c_a$ ) is used by the algorithm to perform the proposed information interaction-enhanced mechanism. Consequently, the parameter tuning process used to achieve a reasonably good performance of IICS can be time consuming. As for the two common parameters of IICS and BCS (i.e.,  $\alpha$  and  $p_a$ ), our current study sets them directly according to the recommendation of Yang and Deb [22, 38]. There may be better value combinations of the three parameters. But their tuning process will no doubt become much more time consuming, and it might also require retuning when the algorithm is applied to solve different optimization problems.

Based on the current study, several future work directions can be pursued. Firstly, a parameter self-learning strategy could be constructed for the proposed IICS algorithm so as to tune the three involved parameters (i.e.,  $\alpha$ ,  $p_a$ , and  $c_a$ ) adaptively. Secondly, there is still room for improvement in the selection strategy of information providers in IICS. The selection of information providers in this article is a kind of blindness; therefore, a more effective selection strategy is worthy to research. Finally, the proposed indirect hybrid modeling method could be also applied to other LF refining processes or other similar complex industrial processes.

**Acknowledgements** This work was supported by the Fundamental Research Funds for the Central Universities (Grant Number N2025032), the Liaoning Provincial Natural Science Foundation (Grant Number 2020-MS-362), and the National Key Research and Development Program of China (Grant Number 2017YFA0700300). The first author would also like to thank China Scholarship Council.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.



## References

- Çamdali Ü, Tunç M (2006) Steady state heat transfer of ladle furnace during steel production process. *J Iron Steel Res Int* 13:18–20
- Volkova O, Janke D (2003) Modelling of temperature distribution in refractory ladle lining for steelmaking. *ISIJ Int* 43:1185–1190
- Wu YJ, Jiang ZH, Jiang MF, Gong W, Zhan DP (2002) Temperature prediction model of molten steel in LF. *J Iron Steel Res* 14:9–12 ((in Chinese))
- Nath NK, Mandal K, Singh AK, Basu B, Bhanu C, Kumar S, Ghosh A (2006) Ladle furnace on-line reckoner for prediction and control of steel temperature and composition. *Ironmak Steelmak* 33:140–150
- Tian HX, Mao ZZ, Wang AN (2009) A new incremental learning modeling method based on multiple models for temperature prediction of molten steel in LF. *ISIJ Int* 49:58–63
- Tian HX, Mao ZZ (2010) An ensemble ELM based on modified AdaBoost. RT algorithm for predicting the temperature of molten steel in ladle furnace. *IEEE Trans Autom Sci Eng* 7:73–80
- Wang XJ, Yuan P, Mao ZZ, You MS (2016) Molten steel temperature prediction model based on bootstrap feature subsets ensemble regression trees. *Knowl-Based Syst* 101:48–59
- Wang X-J, Wang X-Y, Zhang Q, Mao Z-Z (2018) The soft sensor of the molten steel temperature using the modified maximum entropy based pruned bootstrap feature subsets ensemble method. *Chem Eng Sci* 189:401–412
- Tian HX, Mao ZZ, Wang Y (2008) Hybrid modeling of molten steel temperature prediction in LF. *ISIJ Int* 48:58–62
- Lü W, Mao ZZ, Yuan P (2012) Ladle furnace liquid steel temperature prediction model based on optimally pruned bagging. *J Iron Steel Res Int* 19:21–28
- Fu GQ, Liu Q, Wang Z, Chang J, Wang B, Xie FM, Lu XC, Ju QP (2013) Grey box model for predicting the LF end-point temperature of molten steel. *J Univ Sci Technol B* 35:948–954 ((in Chinese))
- Chisari C, Bedon C, Amadio C (2015) Dynamic and static identification of base-isolated bridges using genetic algorithms. *Eng Struct* 102:80–92
- Rahman MA, Anwar S, Izadian A (2016) Electrochemical model parameter identification of a lithium-ion battery using particle swarm optimization method. *J Power Sources* 307:86–97
- Quaranta G, Marano GC, Greco R, Monti G (2014) Parametric identification of seismic isolators using differential evolution and particle swarm optimization. *Appl Soft Comput* 22:458–464
- Bououden S, Chadli M, Karimi HR (2015) An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. *Inform Sci* 299:143–158
- Kang F, Li JJ, Dai JH (2019) Prediction of long-term temperature effect in structural health monitoring of concrete dams using support vector machines with Jaya optimizer and salp swarm algorithms. *Adv Eng Softw* 131:60–76
- Kang F, Li JJ, Ma ZY (2011) Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Inform Sci* 181:3508–3531
- Abd Elazim SM, Ali ES (2016) Optimal power system stabilizers design via cuckoo search algorithm. *Int J Electron Power* 75:99–107
- Schwaab M, Biscoia EC Jr, Monteiro JL, Pinto JC (2008) Non-linear parameter estimation through particle swarm optimization. *Chem Eng Sci* 63:1542–1552
- Lee KH, Kim KW (2015) Performance comparison of particle swarm optimization and genetic algorithm for inverse surface radiation problem. *Int J Heat Mass Transf* 88:330–337
- Xu W, Zhang LB, Gu XS (2011) Soft sensor for ammonia concentration at the ammonia converter outlet based on an improved particle swarm optimization and BP neural network. *Chem Eng Res Des* 89:2102–2109
- Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC 2009). IEEE, Coimbatore, India, pp 210–214
- Shehab M, Khader AT, Al-Betar MA (2017) A survey on applications and variants of the cuckoo search algorithm. *Appl Soft Comput* 61:1041–1059
- Zhu XH, Wang N (2018) Splicing process inspired cuckoo search algorithm based ENNs for modeling FCCU reactor-regenerator system. *Chem Eng J* 354:1018–1031
- Du XJ, Wang JL, Jegatheesan V, Shi GH (2018) Parameter estimation of activated sludge process based on an improved cuckoo search algorithm. *Bioresour Technol* 249:447–456
- Zhao J, Wong PK, Xie ZC, Ma XB, Hua XQ (2019) Design and control of an automotive variable hydraulic damper using cuckoo search optimized pid method. *Int J Autom Technol* 20:51–63
- Meng XJ, Chang JX, Wang XB, Wang YM (2019) Multi-objective hydropower station operation using an improved cuckoo search algorithm. *Energy* 168:425–439
- Mareli M, Twala B (2018) An adaptive cuckoo search algorithm for optimisation. *Appl Comput Inform* 14:107–115
- Yang B, Miao J, Fan ZC, Long J, Liu XH (2018) Modified cuckoo search algorithm for the optimal placement of actuators problem. *Appl Soft Comput* 67:48–60
- Faris H, Sheta A (2016) A comparison between parametric and non-parametric soft computing approaches to model the temperature of a metal cutting tool. *Int J Comput Integr Manuf* 29:1–12
- Zhu XL, Rehman KU, Wang B, Shahzad M (2020) Modern soft-sensing modeling methods for fermentation processes. *Sensors* 20:1771
- Das D, Pal AR, Das AK, Pratihari DK, Roy GG (2020) Nature-inspired optimization algorithm-tuned feed-forward and recurrent neural networks using CFD-based phenomenological model-generated data to model the EBW process. *Arab J Sci Eng* 45:2779–2797
- Civicioglu P, Besdok E (2013) A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artif Intell Rev* 39:315–346
- Bhateja AK, Bhateja A, Chaudhury S, Saxena PK (2015) Cryptanalysis of vigenere cipher using cuckoo search. *Appl Soft Comput* 26:315–324
- Fredman TP, Saxén H (1998) Model for temperature profile estimation in the refractory of a metallurgical ladle. *Metall Mater Trans B* 29:651–659
- Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2:359–366
- Ahmed J, Salam Z (2014) A maximum power point tracking (MPPT) for PV system using Cuckoo Search with partial shading capability. *Appl Energy* 119:118–130
- Yang XS (2014) Nature-inspired optimization algorithms. Elsevier, Oxford
- Sekhar P, Mohanty S (2016) An enhanced cuckoo search algorithm based contingency constrained economic load dispatch for security enhancement. *Int J Elec Power* 75:303–310
- Mantegna RN (1994) Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics* 49:4677–4683
- Yao X, Liu Y, Lin GM (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3:82–102



42. Lim WH, Mat Isa NA (2013) Two-layer particle swarm optimization with intelligent division of labor. *Eng Appl Artif Intel* 26:2327–2348
43. Huang L, Ding S, Yu S, Wang J, Lu K (2016) Chaos-enhanced cuckoo search optimization algorithms for global optimization. *Appl Math Model* 40:3860–3875
44. Naik MK, Panda R (2016) A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition. *Appl Soft Comput* 38:661–675
45. Valian E, Tavakoli S, Mohanna S, Haghi A (2013) Improved cuckoo search for reliability optimization problems. *Comput Ind Eng* 64:459–468
46. Wang LJ, Zhong YW, Yin YL (2016) Nearest neighbour cuckoo search algorithm with probabilistic mutation. *Appl Soft Comput* 49:498–509
47. Chen X, Yu KJ (2019) Hybridizing cuckoo search algorithm with biogeography-based optimization for estimating photovoltaic model parameters. *Sol Energy* 180:192–206
48. Huang ZY, Gao ZZ, Qi L, Duan H (2019) A heterogeneous evolving cuckoo search algorithm for solving large-scale combined heat and power economic dispatch problems. *IEEE Access* 7:111287–111301
49. Noman N, Iba H (2008) Accelerating differential evolution using an adaptive local search. *IEEE Trans Evol Comput* 12:107–125
50. Awad NH, Ali MZ, Suganthan PN, Liang JJ, Qu BY (2016) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization
51. Aydılek İB (2018) A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Appl Soft Comput* 66:232–249
52. Hadi AA, Wagdy A, Jambi K (2018) Single-objective real-parameter optimization: enhanced LSHADE-SPACMA algorithm
53. Zhang G, Shi YH (2018) Hybrid sampling evolution strategy for solving single objective bound constrained problems. *IEEE C Evol Comput* 1–7
54. Gupta S, Deep K (2019) Improved sine cosine algorithm with crossover scheme for global optimization. *Knowl Based Syst* 165:374–406

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.